

UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA  
INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

## Prueba de concepto: Powercenter.

Usos y ejemplos de las herramientas ETL



**Autor:** Sergio Pérez Martín

**Tutora:** Dña. Dolores Cuadra Fernández

## Agradecimientos.

Como todo en esta vida, para todo hay un principio y un final y esta etapa de mi vida ya toca a su fin con este proyecto que entrego hoy. Cuando vine a la presentación de la universidad y del inicio de curso, recuerdo que dijeron muy sabiamente que esto era una carrera de fondo y no una carrera de velocidad, yo hice mío ese lema y muchos años después, tras pasar por ingeniería informática y terminar en la ingeniería técnica, he llegado a la meta y mi carrera termina.

Empecé como un chico recién salido del instituto sin experiencia ni conocimientos y dentro de pocos días seré padre.

En este tiempo he tenido mucha gente a mi alrededor que me ha aportado y ayudado mucho empezando por los profesores de la universidad a los cuales quiero agradecer su esfuerzo y ayuda para sacar lo mejor de mí mismo. Especialmente agradecerle a Dolores Cuadra por su tutelaje y ayuda a lo largo de todo este proyecto.

Acordarme de los compañeros que han ido pasando a mi lado y ayudándome, en las prácticas, por todas esas horas de esfuerzos conjuntos y diferentes puntos de vista que siempre ayudan y te hacen mejorar.

Mencionar a mis compañeros de trabajo que me han ayudado muchísimo y me han ofrecido sus puntos de vista, conocimientos profesionales y todo el apoyo y facilidades que han estado en sus manos. Gracias.

Agradecer a mi familia, a mis padres (Miguel e Isabel) y mi hermana (Lorena) que siempre confiaron en mí y me han animado para que a pesar del esfuerzo mutuo y los años que nos haya tomado llegar a la meta, siempre han estado ahí. Por haberme inculcado el espíritu de esfuerzo y sacrificio. Muchas gracias.

Hacer una especial mención a la persona que inició conmigo una relación personal a la par que comenzaba mi carrera académica, la cual siempre ha confiado en mí y apoyado, la que me ha animado y se ha sentido orgullosa de mí y la que me ha impulsado a dar este último empujón a mi carrera antes de hacerme el mejor regalo de mi vida. Muchas gracias Carmen.

Realizar un último agradecimiento y recordatorio especial a una persona que recientemente nos dejó y que sin ser ingeniero me demostró ser una persona con mucho ingenio, un modelo para no cejar en el empeño de lo que buscas y deseas, superando cualquier adversidad que se te presente en la vida siempre con trabajo y una sonrisa. Gracias a mi abuelo, Mariano.

Por último y aún sin nacer gracias a mi futuro hijo, porque gracias a él conseguí el ánimo y la voluntad para terminar esta carrera de fondo y poner un punto y final a una etapa para comenzar otra mucho más larga y apasionante.

# 1 Índice

1.	Introducción.	11
1.1	Motivación	11
1.2	Objetivos.	11
1.3	Terminología.	12
1.4	Estructura del documento.	14
1.5	Un poco de historia:	15
1.6	Usos de la ETL:	17
1.7	Descripción de la herramienta:	18
1.8	Arquitectura.	22
1.9	Interfaz.	23
1.10	Funcionalidad	37
1.10.1	Funcionalidades del repository manager.	37
1.10.2	Funcionalidades del mapping designer.	55
1.10.3	Funcionalidades del workflow manager.	91
1.10.4	Funcionalidades del workflow monitor.	107
2	Prueba de concepto.	110
2.1	Dominio.	110
2.2	Modelado multidimensional de los datos.	110
2.3	Estructura de las fuentes.	116
2.4	Carga de datos a través de POWERCENTER.	133
2.4.1	Carga de datos para nacimientos y defunciones nacionales.	133
2.4.2	Carga de datos para nacimientos y defunciones nacionales extranjeros.	145
2.4.3	Carga de datos para el censo poblacional.	152
2.4.4	Carga de datos para rentas por sector.	158
2.4.5	Carga de datos para la ocupación de la población.	164
2.5	Análisis de los datos e informes con POWERCENTER.	179
2.5.1	Indicadores concernientes a nacimientos.	179
2.5.2	Indicadores concernientes a defunciones.	183
2.5.3	Indicadores concernientes a la población.	187
3	Conclusiones.	191
3.1	Conclusiones generales.	191

3.2	Líneas de futuro. ....	192
3.3	Bibliografía. ....	192
4	Gestión de proyectos. ....	194
4.1	Planificación. ....	194
4.1.1	Diagrama de Gantt. ....	195
4.2	Presupuesto del proyecto. ....	195
4.2.1	Recursos Hardware. ....	196
4.2.2	Recursos Software.....	197
4.2.3	Recursos humanos. ....	198
4.2.4	Coste del proyecto. ....	199

## Ilustraciones:

Ilustración 1: Repository manager .....	18
Ilustración 2: Designer.....	19
Ilustración 3: Workflow manager.....	20
Ilustración 4: Workflow monitor .....	21
Ilustración 5: Segmentación de la Interfaz del repository manager. ....	23
Ilustración 6: Segmentación de la interfaz del designer .....	25
Ilustración 7: Sistema de ficheros del designer.....	26
Ilustración 8: Analizador de orígenes. ....	27
Ilustración 9: Analizador de destinos. ....	27
Ilustración 10: Desarrollador de transformaciones. ....	28
Ilustración 11: Diseñador de mapplets. ....	29
Ilustración 12: vista del uso o llamada a un mapplet.....	29
Ilustración 13: Diseñador de mappings.....	30
Ilustración 14: Ventanas de logs. ....	30
Ilustración 15: Interfaz workflow manager. ....	31
Ilustración 16: Sistema de ficheros del workflow manager. ....	31
Ilustración 17: Vista sobre el Task developer.....	32
Ilustración 18: Vista sobre el worklet designer.....	32
Ilustración 19: Vista sobre el workflow designer. ....	33
Ilustración 20: Ventana de Logs del workflow manager. ....	33
Ilustración 21: Interfaz workflow monitor, vista de tareas.....	34
Ilustración 22: Vista de las ejecuciones en diagrama de gant. ....	35
Ilustración 23: vista de la ejecución de una sesión. ....	35
Ilustración 24: vista de la ejecución de un Workflow. ....	36
Ilustración 25: Panel del repository manager. ....	37
Ilustración 26: Menú repository.....	38
Ilustración 27: configuración de dominios.....	38



Ilustración 28: Edición de dominios. ....	39
Ilustración 29: Exportar objeto paso 1. ....	40
Ilustración 30: Exportar objeto, paso 2. ....	40
Ilustración 31: Asistente de ayuda import, paso 1. ....	41
Ilustración 32: Asistente de ayuda import, paso 2. ....	42
Ilustración 33: asistente de ayuda import, paso 3. ....	43
Ilustración 34: Asistente ayuda solución de conflictos import. ....	44
Ilustración 35: Asistente de ayuda, resumen de resolución de conflictos. ....	44
Ilustración 36: Proceso de importación. ....	45
Ilustración 37: Menú Folder. ....	45
Ilustración 38: Creación de un directorio en el repositorio. ....	46
Ilustración 39: comparador de directorios, paso 1. ....	46
Ilustración 40: Comparador de directorios, paso 2. ....	47
Ilustración 41: Comparador de directorios, paso 3. ....	47
Ilustración 42: Comparador de directorios, paso 4. ....	48
Ilustración 43: Menú Security. ....	48
Ilustración 44: Modificación de la contraseña de un usuario. ....	49
Ilustración 45: Administrar usuarios. ....	49
Ilustración 46: Crear o editar usuario. ....	50
Ilustración 47: Asignar un usuario a un grupo. ....	51
Ilustración 48: Administrar grupos de usuarios. ....	51
Ilustración 49: Crear/editar grupo. ....	52
Ilustración 50: Listado de privilegios para usuarios y grupos. ....	52
Ilustración 51: Menú de versionado del repositorio. ....	53
Ilustración 52: Menú de análisis. ....	53
Ilustración 53: Menú de utilidades. ....	54
Ilustración 54: Barra de herramientas designer. ....	55
Ilustración 55: Menú edit. ....	55
Ilustración 56: búsqueda en el workspace. ....	56
Ilustración 57: menú layout. ....	56
Ilustración 58: Vista de la ordenación desplegada. ....	57
Ilustración 59: Vista de la ordenación con iconos. ....	57
Ilustración 60: Componentes de un mapping. ....	58
Ilustración 61: ventana auto link. ....	59
Ilustración 62: Componentes enlazados. ....	59
Ilustración 63: Menú Mapping. ....	60
Ilustración 64: ventana de configuración de búsqueda de dependencias. ....	61
Ilustración 65: Tabla de elementos relacionados de un mapeo. ....	61
Ilustración 66: Propagación de atributos. ....	62
Ilustración 67: Comparador de mapeos. ....	63
Ilustración 68: Resultado del comparador de mapeos. ....	64
Ilustración 69: Parámetros y variables. ....	65
Ilustración 70: Plan de carga. ....	66
Ilustración 71: ventana de transformaciones. ....	67
Ilustración 72: Importar tabla de origen. ....	68

Ilustración 73: Vista de orígenes importados. ....	68
Ilustración 74: Importar un origen de fichero, paso 1. ....	69
Ilustración 75: Importar un origen de fichero, paso 2. ....	69
Ilustración 76: Importar un origen de fichero, paso 3. ....	70
Ilustración 77: Importar un origen de fichero, paso 4. ....	70
Ilustración 78: Importar origen desde xml, paso 1. ....	71
Ilustración 79: Importar origen desde xml, paso 2. ....	71
Ilustración 80: Importar origen desde xml, paso 3. ....	72
Ilustración 81: Definición del origen XML. ....	72
Ilustración 82: Creación de un origen. ....	73
Ilustración 83: Configuración de un origen. ....	73
Ilustración 84: Crear campos en un origen. ....	74
Ilustración 85: Menú destinos. ....	76
Ilustración 86: Menú transformación. ....	77
Ilustración 87: Importar procedimientos. ....	79
Ilustración 88: Procedimientos importados. ....	80
Ilustración 89: Creación de transformaciones. ....	81
Ilustración 90: Nueva transformación. ....	81
Ilustración 91: Definición de campos entrada y salida. ....	82
Ilustración 92: Definición del código java a ejecutar. ....	82
Ilustración 93: Crear un Mapplet. ....	83
Ilustración 94: Componentes disponibles para el desarrollador. ....	84
Ilustración 95: Ejemplo de mapplet. ....	84
Ilustración 96: Desarrollo de un mapeo, orígenes. ....	85
Ilustración 97: Desarrollo de un mapeo, expresiones. ....	86
Ilustración 98: Diseño de un mapeo, clasificación de campos en una expresión. ....	86
Ilustración 99: Diseño de un mapeo, operaciones sobre campos. ....	87
Ilustración 100: Diseño de un mapeo, filtros. ....	87
Ilustración 101: Diseño de un mapeo, filtros. ....	88
Ilustración 102: Diseño de un mapeo, lógica del mapplet. ....	89
Ilustración 103: Diseño de un mapeo. Configuración de un componente de ordenación. ....	89
Ilustración 104: Diseño de un mapeo, utilización de la expresión. ....	90
Ilustración 105: Diseño de un mapeo, configuración del agregador. ....	90
Ilustración 106: Diseño de un mapeo, secuenciador y destino. ....	91
Ilustración 107: icono y menú de creación de sesiones. ....	92
Ilustración 108: Ventana de selección de mapeo. ....	92
Ilustración 109: icono de sesión. ....	92
Ilustración 110: pestaña de propiedades. ....	93
Ilustración 111: pestaña de opciones de configuración del objeto sesión. ....	94
Ilustración 112: configuración de elementos de la sesión. Orígenes. ....	95
Ilustración 113: configuración de elementos de la sesión. Destinos. ....	96
Ilustración 114: explorador de conexiones a BBDD. ....	97
Ilustración 115: Propiedades de los destinos. ....	97
Ilustración 116: Pestaña Components. ....	98
Ilustración 117: Crear un command. ....	98

Ilustración 118: Ventana de configuración y creación de comandos. ....	99
Ilustración 119: Ventana de edición de comandos. ....	99
Ilustración 120: pestaña de programación. ....	100
Ilustración 121: Ventana de programación de ejecuciones del wf. ....	101
Ilustración 122: workflow programado para ejecución horaria. ....	102
Ilustración 123: Ventana de creación y configuración de workflow. ....	103
Ilustración 124: componentes utilizables en el workflow. ....	104
Ilustración 125: Estructura general del workflow, inserción de la primera sesión. ....	104
Ilustración 126: filtros en las líneas de conexión. ....	105
Ilustración 127: Estructura general del workflow, worklet. ....	105
Ilustración 128: diseño del workflow, contenido del worklet. ....	106
Ilustración 129: Estructura general del workflow, inserción del command. ....	106
Ilustración 130: Opciones sobre un workflow. ....	107
Ilustración 131: workflow log. ....	108
Ilustración 132: Modelo de datos. ....	111
Ilustración 133: Proceso de carga visto globalmente. ....	133
Ilustración 134: Lectura de ficheros. ....	133
Ilustración 135: Apariencia del componente expresión y pantalla de configuración. ....	134
Ilustración 136: Obtención y limpieza de la variable provincia. ....	135
Ilustración 137: Componente filtro y configuración del filtro. ....	137
Ilustración 138: Componente router y su configuración. ....	137
Ilustración 139: Entrada de los registros a los mapplets en el flujo principal de carga. ....	138
Ilustración 140: Entrada de datos en el mapplet. ....	139
Ilustración 141: Bifurcación de datos en meses y configuración del router. ....	139
Ilustración 142: Configuración de las expresiones mensuales y funciones aplicadas en el cálculo del nuevo campo. ....	140
Ilustración 143: Unificación de flujos y salida del mapplet. ....	140
Ilustración 144: Configuración del componente UNION. ....	141
Ilustración 145: Creación y configuración del nuevo campo fecha. ....	141
Ilustración 146: Finalización del proceso de carga. ....	142
Ilustración 147: Vista general del workflow. ....	142
Ilustración 148: Configuración de los logs de sesión. ....	143
Ilustración 149: Configuración de orígenes en la sesión. ....	144
Ilustración 150: Configuración de destinos en la sesión. ....	145
Ilustración 151: Lectura del fichero de entrada. ....	146
Ilustración 152: Configuración de la expresión y función de cambio de tipos. ....	146
Ilustración 153: Configuración del router. ....	147
Ilustración 154: Router para la bifurcación mensual para mujeres. ....	148
Ilustración 155: Configuración de la expresión mensual y expresión para la fecha. ....	148
Ilustración 156: Definición de parámetros de entrada. ....	149
Ilustración 157: unión de los meses en el componente UNION y configuración del mismo. ....	149
Ilustración 158: Unificación de flujos por sexo. ....	150
Ilustración 159: Bifurcación por nacimiento o defunción e inserción en tabla. ....	150
Ilustración 160: Configuración del router de nacimientos y defunciones. ....	151
Ilustración 161: Vista general del workflow. ....	151

Ilustración 162: Vista general del mapping de censo poblacional.....	152
Ilustración 163: Configuración de la expresión de suma de poblaciones de municipios. ....	153
Ilustración 164: Router y configuración de los grupos del mismo.....	153
Ilustración 165: Configuración de las expresiones de censos por municipios.....	154
Ilustración 166: Inserción en las tablas de destino. ....	155
Ilustración 167: Workflow PFC_CENSO.....	155
Ilustración 168: Configuración del fichero de entrada. ....	156
Ilustración 169: Configuración de las tablas de salida. ....	157
Ilustración 170: Vista general del mapping PFC_RENT_SEC. ....	158
Ilustración 171: Expresión inicial, cambio de tipos y calculo de provincia y sector.....	159
Ilustración 172: Filtro y configuración del filtro.....	160
Ilustración 173: Router Sexo y configuración. ....	161
Ilustración 174: Router de años y configuración. ....	161
Ilustración 175: Expresión creación campo sexo. ....	162
Ilustración 177: workflow PFC_RENT_SEC. ....	162
Ilustración 178: Configuración del fichero de logs de sesión.....	163
Ilustración 179: Configuración de un fichero de entrada directo. ....	163
Ilustración 180: Configuración de la tabla de destino PFC_RENT_SEC. ....	164
Ilustración 181: Vista general del proceso de carga m_PFC_pob_activa_04. ....	165
Ilustración 182: Lectura de datos y primera expresión de transformación.....	165
Ilustración 183: Router para tipo de estado. ....	166
Ilustración 184: Expresión tipo estado y configuración. ....	166
Ilustración 185: Llamada a la lookup.....	167
Ilustración 186: Configuración de campos de la lookup. ....	168
Ilustración 187: Filtro de condición de la lookup. ....	168
Ilustración 188: Unificación de flujos. ....	169
Ilustración 189: Cálculo del año e inserción en tabla.....	170
Ilustración 190: Aspecto general del mapeo.....	171
Ilustración 191: Configuración de la expresión inicial.....	172
Ilustración 192: Vista general del mapplet mp_convertir_filas_pob_act.....	172
Ilustración 193: Configuración del router para el tipo de estado. ....	173
Ilustración 194: Configuración lookup desconectada.....	173
Ilustración 195: Invocación a la lookup desde la expresión.....	174
Ilustración 196: lookup conectada. ....	174
Ilustración 197: Configuración de la lookup conectada.....	175
Ilustración 198: Unión de los flujos de datos y salida del mapplet.....	175
Ilustración 199: Expresión de cálculo del año e inserción en bbdd. ....	176
Ilustración 200: workflow con sesiones en serie. ....	176
Ilustración 201: workflow con sesiones en paralelo. ....	176
Ilustración 202: configuración de los logs de sesión.....	177
Ilustración 203: Configuración del fichero de entrada. ....	177
Ilustración 204: Configuración de las salidas. ....	178
Ilustración 205: Configuración de la lookup.....	178
Ilustración 206: Diagrama de Gantt. ....	195

## Tablas:

Tabla 1: Representación de los valores de los registros antes de la agregación. ....	89
Tabla 2: registro una vez se han agregado sus valores. ....	90
Tabla 3: PFC_MES. ....	112
Tabla 4: PFC_SEXO. ....	112
Tabla 5: PFC_MUNICIPIO_PROVINCIA. ....	112
Tabla 6: PFC_PROVINCIA. ....	112
Tabla 7: PFC_TIPO_ESTADO. ....	113
Tabla 8: PFC_SECTOR. ....	113
Tabla 9: PFC_ANYO. ....	113
Tabla 10: PFC_COMUNIDAD. ....	113
Tabla 11: PFC_NACIMIENTOS. ....	113
Tabla 12: PFC_DEFUNCIONES. ....	114
Tabla 13: PFC_DEFUNCIONES_EXT. ....	114
Tabla 14: PFC_NACIMIENTOS_EXT. ....	114
Tabla 15: PFC_CENSO. ....	114
Tabla 16: PFC_GANANCIAS. ....	114
Tabla 17: PFC_POB_OCUPACION. ....	115
Tabla 18: Formato de ficheros de nacimientos y defunciones nacionales. ....	118
Tabla 19: Formato de ficheros de nacimientos y defunciones pre procesados. ....	120
Tabla 20: Formato del fichero de nacimientos y defunciones de extranjeros. ....	122
Tabla 21: Formato del fichero de nacimientos y defunciones extranjeros pre procesado. ....	123
Tabla 22: Formato del fichero de censos. ....	124
Tabla 23: Formato del fichero de censos pre procesado. ....	125
Tabla 24: Fichero de ocupación de la población año 2004. ....	126
Tabla 25: Fichero de ocupación de la población pre procesado. ....	127
Tabla 26: Ficheros de ocupación de la población para los años 2005 al 2013. ....	128
Tabla 27: Ficheros de ocupación de la población para los años 2005 al 2013 pre procesados. ....	129
Tabla 28: Fichero de rentas por sector y comunidad. ....	131
Tabla 29: Fichero de rentas por sector y comunidad pre procesado. ....	132
Tabla 30: Registro de provincia. ....	135
Tabla 31: Registro con datos mensuales. ....	135
Tabla 32: Tasa de natalidad bruta por provincia y municipio. ....	179
Tabla 33: Tasa de natalidad bruta para extranjeros. ....	180
Tabla 34: Ratio de masculinidad y feminidad. ....	182
Tabla 35: Tasa de defunciones bruta por provincia y municipio. ....	183
Tabla 36: Tasa de defunción bruta de extranjeros por provincia. ....	184
Tabla 37: Ratio de masculinidad/feminidad a la defunción para extranjeros. ....	186
Tabla 38: Saldo vegetativo. ....	188
Tabla 39: Relación de parados y número de nacimientos. ....	188
Tabla 40: Relación salarial entre hombres y mujeres por comunidad y sector. ....	189
Tabla 41: Lista de tareas. ....	194
Tabla 42: Coste de recursos humanos. ....	198
Tabla 43: Tabla de coste de recursos humanos. ....	199
Tabla 44: Tabla de costes de hardware. ....	200

Tabla 45: Tabla de costes de software. .... 200

**Ecuaciones:**

Ecuación 1: Tasa de natalidad bruta por provincia. .... 179

Ecuación 2: Ratio de masculinidad/feminidad al nacimiento. .... 181

Ecuación 3: Tasa de mortalidad bruta por provincia. .... 183

Ecuación 4: Ratio de masculinidad/feminidad a la defunción por provincia. .... 185

Ecuación 5: Saldo vegetativo por provincia. .... 187

## **1. Introducción.**

En el mundo empresarial a lo largo de los últimos años ha surgido la necesidad de integrar las distintas fuentes de información acerca de un entorno concreto, esta información estaba disponible en distintos contenedores de información no unificada. El problema que se encontraban dichas entidades, sobre todo las grandes empresas, es que esta información la tenían disgregada en distintas fuentes, lo cual hacía que su procesamiento y análisis para extraer alguna conclusión a partir de ellos fuera muy costosa y lenta.

Como solución a la integración y unificación de estos datos surge el concepto de ETL (en español extracción, transformación y carga) y las distintas herramientas que realizan dichas tareas. Estas herramientas son un paso previo para poder utilizar las herramientas de Business Intelligent (BI) que proporcionarán a su destinatario, informes o estrategias que pueden beneficiar la toma de decisiones en su empresa, recogiendo la información más relevante de distintas fuentes e incluso siendo éstas, fuentes heterogéneas.

Las herramientas ETL ayudan a realizar la unificación e integración de los datos contenidos en estas fuentes diversas y realizan un pre procesamiento y carga de todos esos datos en un único sistema o base de datos, para posteriormente ser utilizados en la toma de decisiones de una empresa u organización de acuerdo al análisis de los mismos.

### **1.1 Motivación**

El motivo de este proyecto es mostrar las diversas funcionalidades y ayudas que pueden ofrecer las ETLs en el mundo de los procesos de cargas y las BBDD. Se intentará mostrar la versatilidad y la potencia de dichas herramientas a través de un ejemplo práctico de carga sobre el cual intentaremos obtener después alguna información curiosa tras haber pre procesado y cargado en BBDD los datos correspondientes.

De esta manera, quería ofrecer una imagen de lo que puede ser un día corriente en un puesto de trabajo como el que vengo desarrollando yo a diario con estas herramientas, mostrando su eficiencia y dejando un documento o guía de uso acerca de la misma ya que al ser una herramienta de pago no suele haber mucha documentación al respecto de ella en internet.

El caso práctico que elegiremos para ilustrar el proceso de carga y los pasos para la utilización de una ETL como la que referimos será el estudio de distintos aspectos de la población española y algunos indicadores sociales, de los cuales esperamos poder obtener conclusiones acerca de como ha ido evolucionando la población en distintos aspectos y como afectan ciertos problemas sociales como la discriminación de sexos en el trabajo y la crisis económica a las distintas variables que estudiaremos.

### **1.2 Objetivos.**

En este documento se pretende ofrecer al lector una perspectiva sobre la versatilidad y utilidad de las herramientas ETL. En concreto la herramienta objeto de estudio que en este caso será la ETL de pago POWERCENTER versión 8.1.1.

Para ello realizaremos un recorrido sobre distintos aspectos de la herramienta, los tres aspectos fundamentales a estudio serán:

- **Arquitectura:** en este punto vamos a explicar la arquitectura que tiene una instalación de PWC.
- **Interfaz:** donde analizaremos las interfaces de las distintas aplicaciones de las que se compone la ETL para ofrecer todas sus funcionalidades.
- **Funcionalidad:** enumeraremos y describiremos las funcionalidades más usuales y potentes de las distintas aplicaciones que componen esta ETL.
- **Ejemplo de carga:** Realizaremos un ejemplo de carga bajo unas circunstancias concretas especificadas en el caso de ejemplo.

A la finalización de este documento, se pretende haber mostrado más en profundidad el aspecto, funcionalidad y funcionamiento de esta herramienta ETL y poder dar una visión al lector de sus posibilidades y utilidades en sus proyectos, así como la relevancia que tienen estas herramientas en todos los proyectos de BI.

Por otro lado mediante el caso práctico de la prueba de concepto ilustraremos un proceso de carga y procesamiento de datos e intentaremos obtener ejemplos y conclusiones acerca de los indicadores demográficos básicos y algunos resultados curiosos que estaban “ocultos” entre los datos, que se deriven del estudio de los datos una vez procesados.

### 1.3 Terminología.

**BI:** Business Intelligence.

**ETL:** “Extract, transformation Load” en español Extracción, transformación y carga.

**BBDD:** Base de datos.

**Data mart:** Almacén de datos de un área específica dentro del negocio.

**Data warehouse:** Almacenes de datos orientados a un determinado ámbito y que contienen una gran cantidad de datos.

**XML:** “eXtensible Markup Language” en español lenguaje de marcas extensible. Lenguaje que permite definir la gramática de lenguajes específicos para estructurar grandes documentos.

**Disparadores:** También llamados triggers, son procedimientos que se ejecutan en la base de datos cuando se cumple una condición establecida.

**Background:** Se utiliza para nombrar aquellos procesos que se ejecutan en segundo plano, sin que a priori el usuario sea consciente de su ejecución.

**Lookups:** Son consultas que se realizan sobre las tablas y que sirven como apoyo para una comprobación determinada.

**PWC:** Power Center.

**SO:** Sistema Operativo.

**FTP:** “File Transfer Protocol” es un protocolo de transferencia de ficheros.

**SQL:** lenguaje declarativo de acceso a bbdd relacionales.



**SH:** Shell Script, es un archivo de órdenes o comandos para un sistema operativo, en este caso para UNIX.

## 1.4 Estructura del documento.

Este documento constará de cinco capítulos en los cuales se irá desarrollando la explicación de qué es una ETL como Powercenter y cuáles son sus funcionalidades y utilidades.

El desglose de capítulos será el siguiente:

- **Capítulo 1:** Introducción. En este capítulo se hará una breve introducción de lo que son las ETLs, cuales son los objetivos de este documento y los términos relacionados con las ETL y las BBDD.
- **Capítulo 2:** Recorrido por la herramienta Powercenter. Aquí se irá detallando cuales son las aplicaciones que componen el paquete de herramientas de Powercenter, las funcionalidades, usos, interfaces y varios aspectos relacionados con la herramienta.
- **Capítulo 3:** Prueba de concepto. En este apartado realizaremos con todo detalle un proceso de carga con la herramienta y analizaremos sus orígenes, destinos resultados y todo el proceso de carga paso por paso.
- **Capítulo 4:** Conclusiones. Aquí el autor ofrecerá las conclusiones sobre el estudio de la prueba de concepto realizada, expresando su opinión partiendo de las premisas que suponía al comienzo del documento y en base a los resultados obtenidos.
- **Capítulo 5:** Gestión del proyecto. En este apartado se realizará un presupuesto sobre lo que costaría un proyecto realizado con esta herramienta y la planificación de tiempos y costes del mismo.

## 2 Un recorrido por la herramienta: Powercenter.

En esta sección se describirá la herramienta Powercenter, pero antes de adéntranos en ella presentaremos una visión de los conceptos que sustentan esta tecnología.

### 2.1 Un poco de historia:

Para familiarizarnos con este concepto analicemos su definición. [ETL](#) (son las siglas de Extraer, Transformar y Cargar que viene del inglés Load) es el proceso que permite mover datos desde distintas fuentes, realizar transformaciones, limpieza o reformato de datos y su posterior carga en distintos destinos, ya sean bases de datos ([bbdd](#)), [data mart](#), [data warehouse](#) o ficheros.

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados, es decir, sistemas que existían antes de la utilización de la herramienta y que esta permite usarla para la recolección e integración de sus datos en el nuevo entorno.

El proceso de **extracción** consiste en obtener los datos desde distintos sistemas de origen. Estos sistemas no tienen por qué estar ubicados en la misma máquina e incluso pueden tener una naturaleza diferente, es decir ser sistemas heterogéneos, como pudieran ser una bbdd de cualquier tipo, ficheros planos o ficheros [xml](#). La extracción convierte los datos a un formato preparado, homogéneo. Para ese proceso de extracción todos los datos obtenidos tendrán una estructura la cual siempre será la misma y sobre la cual posteriormente se realizarán las transformaciones necesarias.

Una cuestión importante a tener en cuenta es que el proceso de extracción cause el menor impacto posible sobre los sistemas de origen de los cuales se extrae los datos, ya que estos sistemas son de uso cotidiano para los usuarios y podría llegar a colapsarlos o saturarlos, haciendo que su uso se ralentice y afecte a su uso normal. Por ello, se suele usar la estrategia de programar la extracción de datos en horarios o días en los que las máquinas no tengan mucho uso o incluso utilizar alguna máquina de apoyo intermedia en la cual dejar la información para que podamos realizar dicha extracción sin ningún tipo de perjuicio.

La fase de **transformación** aplica distintas funciones sobre los datos anteriormente extraídos y los prepara y formatea para su posterior carga en el destino. Las fuentes de datos requerirán en algunos casos pequeñas manipulaciones y en otros casos, de transformaciones más complejas.

El programador puede utilizar desde la herramienta todas las funciones propias una bbdd (como `substr`, `decode`, `to_date`, etc) sobre los datos que recibe como entrada y que se procesarán con ella. Además permite realizar funciones más complejas, propias de la herramienta como serían: enrutadores (que funcionalmente cumpliría el papel de direccionar en base a una condición, el flujo y procesamiento que recibirán los datos, como si fuera un IF-ELSE en programación), filtros de datos, joins( cruces entre tablas), agregadores, inserción de métodos programados en java u otros lenguajes para realizar algún tipo de función especial, etc.

Por último la fase de **carga** es en la cual los datos ya transformados y procesados van a ser insertados en los distintos sistemas de destino. Dichos sistemas pueden ser variados: ficheros planos, bbdd, data warehouse, data mart, etc.

La información puede ser insertada de varias maneras, sobrescribiendo los datos con datos nuevos o manteniendo un historial de los registros a lo largo del tiempo como se haría en un data warehouse.

El proceso puede tener como destino distintas opciones, dependiendo de las necesidades del proceso en cuestión, se podrá escribir en ficheros (.txt, .xml, .tmp, etc) o en una bbdd en la cual se quiera contener la información extraída. Al definir una bbdd y una tabla de la misma como destino, si realizamos una inserción en esta tabla, se aplicaran sobre los datos que vayamos a insertar, todas las restricciones o disparadores que haya definidos sobre dicha tabla. De esta manera si el destino es una tabla de una bbdd conseguimos una integridad en los datos y asegurarnos que estos son válidos y sin errores.

## 2.2 Usos de la ETL:

Como ya comentamos en la introducción de este trabajo el marco sobre el cual se aplican este tipo de herramientas son el ámbito de Business Intelligent en el cual será muy importante tener una colección de datos bien estructurados y preparados para la elaboración de informes que puedan utilizarse para la toma de decisiones en los distintos entornos de una empresa. El uso de la herramienta ETL no tiene mucho sentido si no es para una posterior explotación de los datos obtenidos por una herramienta BI.

Tenemos que entender las herramientas ETL como un paso muy importante y previo en cualquier proyecto [BI](#) (Business Intelligence) ya que serán las encargadas de transformar la información en bruto de los distintos orígenes (pudiendo ser estos, ficheros que haya obtenido como salida de otra aplicación, ficheros planos sin algún tipo de formato, tablas de una bbdd) en una información estructurada, agrupada o agregada con un formato concreto y predefinido, que permitirá al usuario final mediante el uso de las herramientas BI obtener las conclusiones y conocimiento acerca de las medidas a tomar de manera más rápida y efectiva.

Este tipo de herramientas no tienen un resultado visible para los usuarios finales de BI ya que ejecutan procesos que se realizan en “[background](#)” pero el tipo de operaciones que realizan puede llevarse hasta un 70% de las necesidades y recursos de un desarrollo BI. Por eso el diseño de la carga es vital para llevar a una situación ventajosa a una organización.

Las herramientas ETL tienen su principal uso como herramientas de interconexión de distintas fuentes de información para su posterior integración en otros sistemas de datos, en los cuales estos datos ya estarán formateados, preparados y listos. Estos, se insertaran para una posterior explotación de los mismos mediante módulos de informes y minería de datos, ya que el uso final es el estudio de tendencias que puedan maximizar beneficio, reducir costes y definir líneas de actuación de cara al posicionamiento del cliente en el sector.

Otro de los usos de este tipo de herramientas es el de suplir las necesidades de pequeñas aplicaciones de modo que garanticen la disponibilidad de la información cuando los usuarios de la misma trabajen con ella. Aquí, lo que se intenta es que a partir de la información que hay en distintos orígenes sin depurar se ajuste a las necesidades que requiera la aplicación en concreto.

Otros usos que podemos dar a las herramientas ETL son:

- Conectividad: la posibilidad de conectarse a distintos orígenes, bases de datos relacionales o no, distintos formatos de archivos, xml, otras herramientas empresariales...
- Capacidad de metadatos y modelado de datos: recupera modelos de datos de los respectivos orígenes, mapeo del modelo físico al lógico, por ejemplo, con ficheros planos, sincronización de cambios en distintos componentes de la herramienta.
- Capacidad de diseño y entorno de desarrollo: establece una representación grafica de los objetos del repositorio y los flujos de datos que se establecen entre los distintos componentes.
- Transformación de datos: transformar los datos desde las operaciones más simples a las operaciones más complejas como sumatorios, agregaciones o [lookups](#).

## 2.3 Descripción de la herramienta:

La herramienta ETL que vamos a analizar como anteriormente hemos mencionado será la aplicación de pago POWER CENTER 8.1.1.

Esta herramienta consta de 4 diferentes aplicaciones que controlan y desarrollan distintos aspectos relativos a los procesos de carga de la ETL. Las cuatro aplicaciones que la componen son las siguientes:

- **Repository manager:** Esta aplicación se encarga de la gestión y administración de los repositorios definidos en la ETL. (Repositorio en este caso es un almacén que contiene toda la información de los procesos ETL que tenemos creados). Nos ofrece las posibilidades de crear, mover, eliminar directorios del repositorio, adjudicar permisos sobre los distintos directorios, configurar y crear dominios y repositorios, realizar versionado de código, importar y exportar procedimientos de carga, etc.

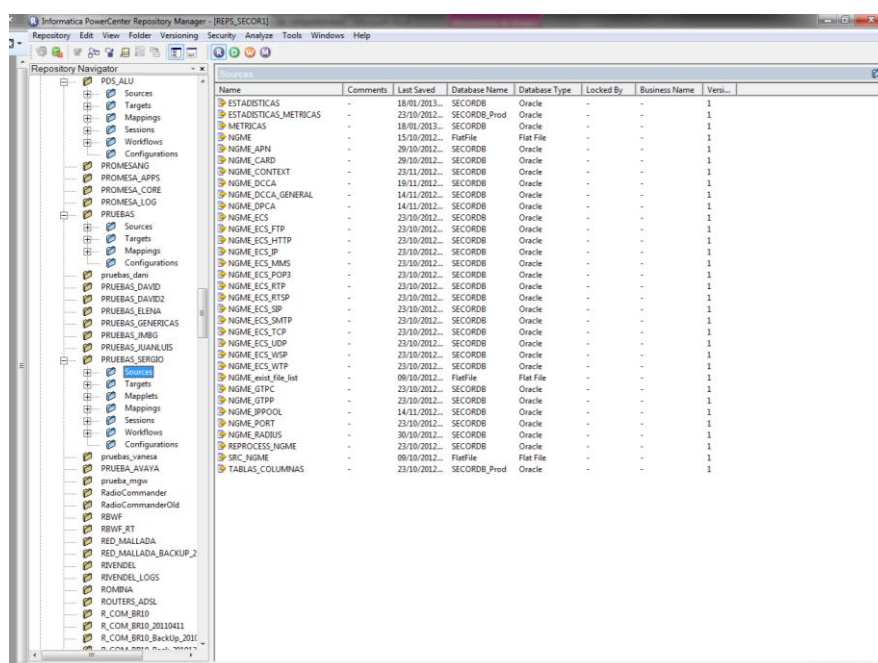


Ilustración 1: Repository manager

- **Designer:** Es el encargado de realizar la lógica de negocio. A más bajo nivel, indica el flujo que seguirán los datos. Aquí el desarrollador realiza las transformaciones necesarias e indica las relaciones de campos entre el origen y el destino así como las funciones que se aplicaran sobre los datos y que procedimientos seguiremos con esos en su llegada a destino informado de si se borrraran, actualizaran o insertaran dichos datos.

Para establecer un símil con la programación más tradicional, diríamos que esta aplicación crea lo que sería una “clase” en un lenguaje orientado a objetos. Definimos que se harán con los datos, transformaciones, en que momento insertaremos, de donde obtendremos los datos, pero solo establecemos una lógica, un patrón, no hay unos datos concretos de momento con los que vayamos a trabajar, todo esto es un diseño de manera abstracta.

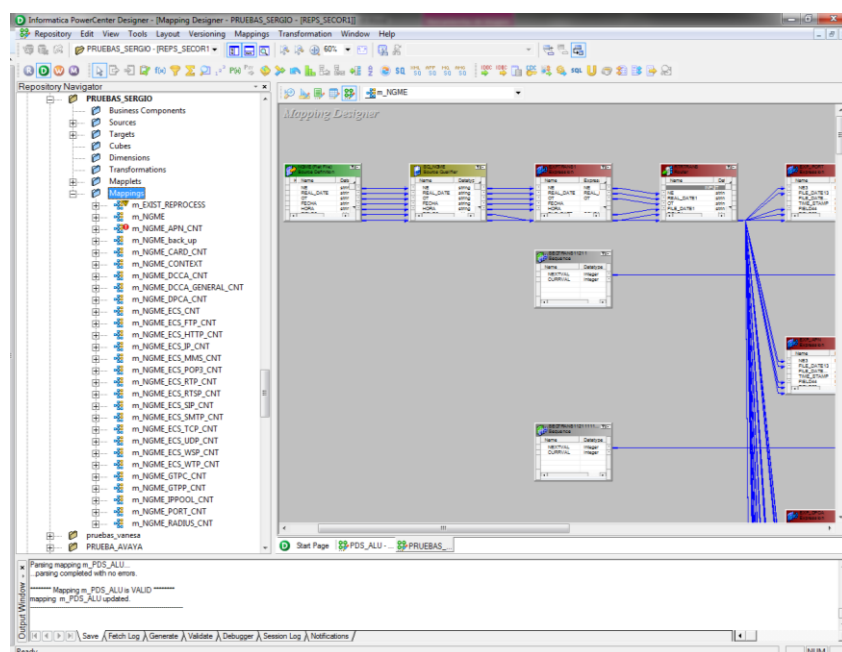


Ilustración 2: Designer

- **Workflow manager:** Esta aplicación se encarga entre otras cosas de la lógica a alto nivel, aquí los procesos pueden subdividirse en distintos subprocessos que se encarguen de pequeñas cargas e ir realizando comprobaciones entre los subprocessos para llegar a realizar una lógica más compleja.

También podemos programar las ejecuciones de las tareas para las horas y periodicidades que se estipulen.

Continuando con la similitud con un lenguaje de programación, podríamos decir que esta aplicación crea el “main” e instancia los objetos de las clases en las que se basa en el momento de la ejecución.

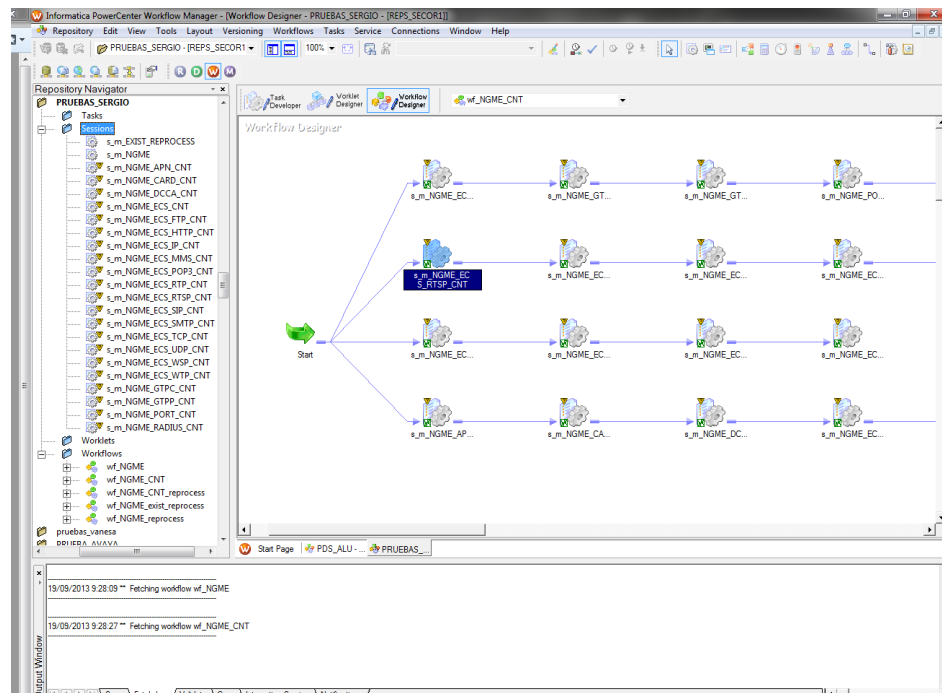


Ilustración 3: Workflow manager



- **Workflow monitor:** Esta última aplicación muestra al desarrollador todas las tareas programadas así como la situación de los procesos de ejecución y los logs de como se ha desarrollado el proceso lanzado. Además nos muestra en detalle cuantos datos se han insertado y leído de cada origen y destino.

Desde ella podemos programar y desprogramar los procesos, los cuales hemos definido su programación en el Workflow manager anteriormente explicado.

Por finalizar los símiles con los métodos de programación tradicional, podríamos explicar esta aplicación como la consola o servidor en la cual se ejecutan los programas y nos va mostrando los resultados y los logs.

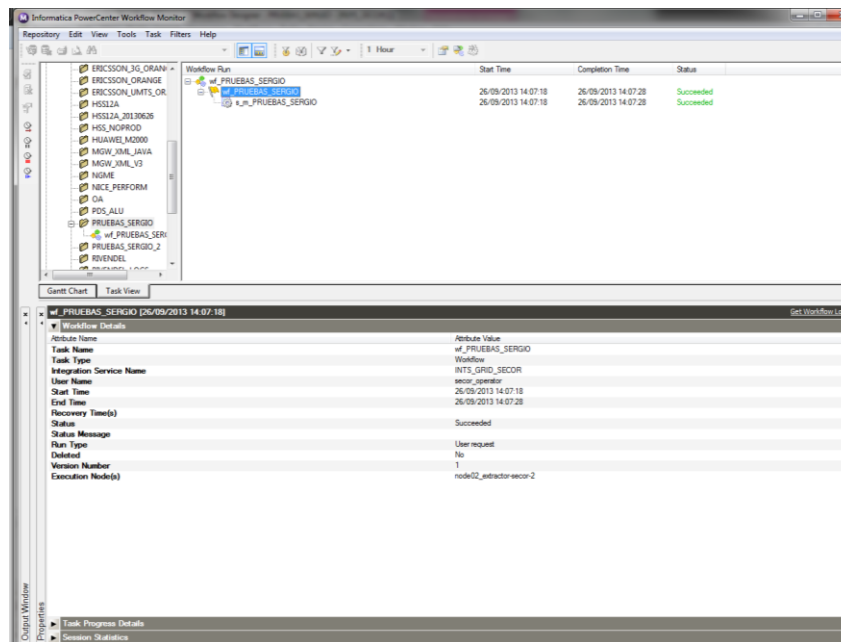


Ilustración 4: Workflow monitor

## 2.4 Arquitectura.

Powercenter es un aplicación cliente /servidor. Este proyecto se centrará sólo en la parte cliente de esta herramienta.

Sobre la parte de servidor sí puedo comentar que tiene una base de datos que tendrá el inventario de todo lo que en la aplicación se genere y necesite ser almacenado, además de parámetros de configuración propios de la aplicación.

Es necesario instalarse la aplicación cliente, que se conectará al servidor y a los repositorios que este tenga configurados. También habrá que instalar una o varias bases de datos en la que se almacenen los datos extraídos de las distintas fuentes.

La conexión desde el cliente a las distintas bases de datos que contengan los datos de origen se realizan mediante ODBC, para cada distinta BBDD a la cual se quiera conectar habrá que crear la pertinente entrada en el fichero odbc.ini y habrá que tener el conector adecuado para ella. Se puede conectar a las bbdd que se deseen y de cualquier tipo siempre que se disponga de dicho conector.

Por otro lado, también puede conectarse a distintas máquinas para recoger los datos de un fichero en lugar de una tabla de bbdd, para conectarse a estas máquinas la aplicación dispone de varios componentes que pueden realizar este tipo de funciones como conexiones ftp, sftp o ssh. Normalmente la conexión a otras máquinas para la obtención de los ficheros de origen se realiza en un script previo a la invocación del proceso de Powercenter y este script prepara el fichero y lo deja en un directorio que será el que recoja Powercenter para su procesado.

La escritura de datos en el destino también puede realizarse sobre tablas de bbdd o sobre ficheros. Normalmente esta bbdd de destino es una bbdd exclusiva para recibir los datos extraídos y después que éstos puedan ser explotados por otra herramienta BI, pero pueden escribirse los datos sobre cualquier bbdd que tengamos configurados en el fichero odbc.ini y que tengamos configurada su conexión en la aplicación.

En caso de querer escribir a un fichero, este se escribirá en el directorio de la máquina en la que tenga configurado la aplicación servidora que está el sistema de ficheros de la aplicación.

## 2.5 Interfaz.

En este punto del documento vamos a proceder a mostrar y describir las interfaces de cada aplicación que tenemos a nuestra disposición en la herramienta. Procederemos a mostrar las interfaces de los distintos entornos de trabajo para cada una de ellas: repository manager, designer, Workflow manager y Workflow monitor.

En cada aplicación podremos definir nuestros escritorios de trabajo de acuerdo a nuestros gustos y añadiendo o eliminando barras de herramientas u ocultando o mostrando vistas podremos llegar a dejar la apariencia de trabajo que deseemos.

Empezaremos a detallar la interfaz de la aplicación **Repository manager**:

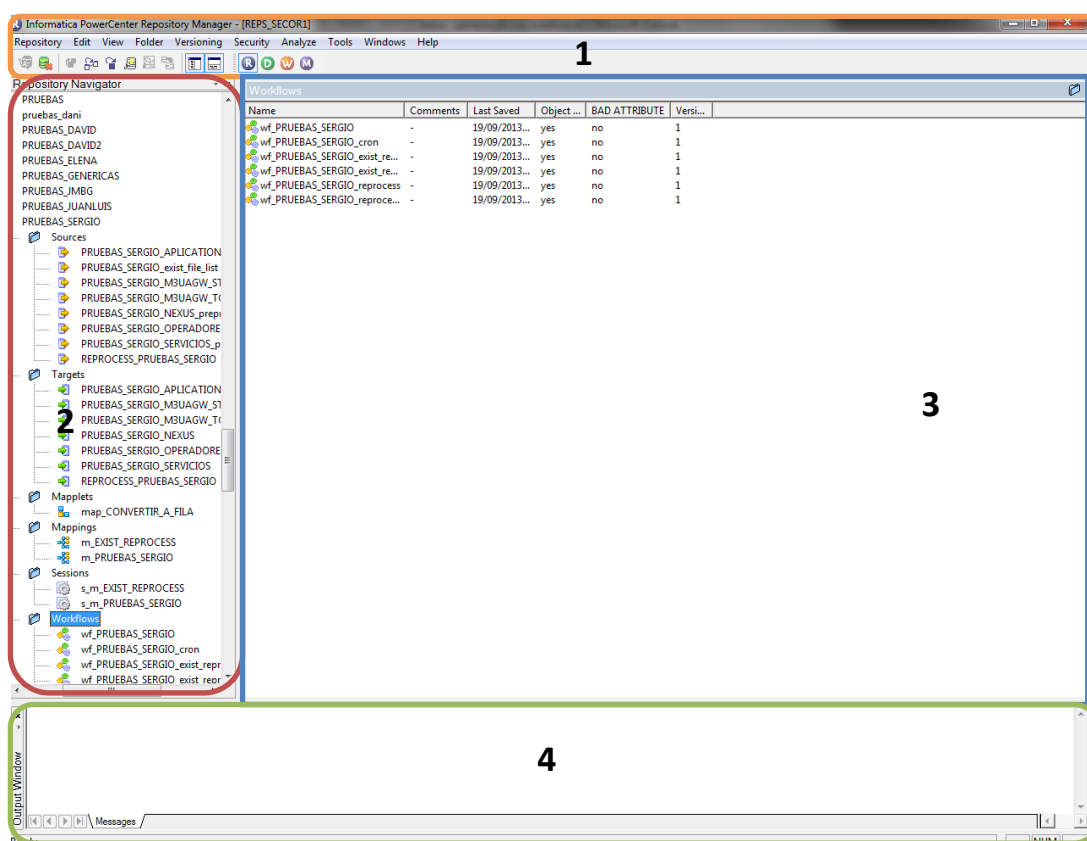


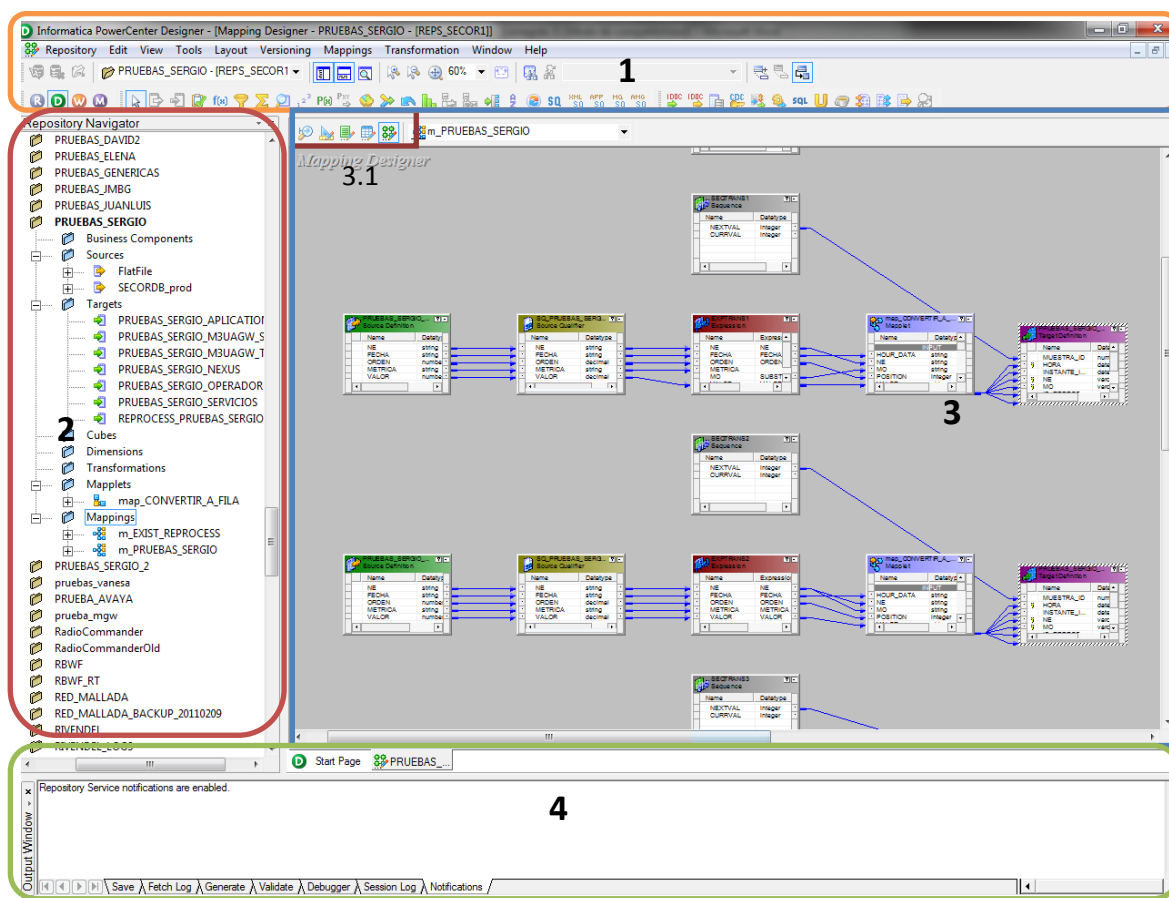
Ilustración 5: Segmentación de la Interfaz del repository manager.

El aspecto o interfaz de esta aplicación se puede dividir en 4 apartados:

- En el primer recuadro marcado de la imagen, se sitúan las barras de herramientas, por un lado están los clásicos desplegados que contienen todas las opciones de la aplicación y por otro están las barras de herramientas configurables que contienen por temática o por cada distinta opción del menú desplegable los iconos de las funcionalidades de manera directa. En este cuadrante el usuario podrá añadir y eliminar a su gusto todas las opciones que permita la aplicación para tener más a su alcance las funciones más comunes en su trabajo diario, podrá añadir barras de herramientas pre configuradas y dentro de estas añadir o eliminar iconos para adecuar a sus necesidades el entorno de trabajo.

- En el segundo cuadrante, se encuentra el navegador o explorador del repositorio, aquí se muestra al desarrollador todo el sistema de ficheros que tiene el repositorio y puede desplazarse entre las distintas carpetas y abrirlas o cerrarlas (realmente lo que hace es conectarse a ese directorio en concreto del repositorio el resto están cerrados, es como un check in de un subversión) para ver su contenido y proceder a su edición que en cada aplicación contendrá distintos elementos que después formaran un todo que será en proceso ETL concreto que se haya desarrollado. En esta aplicación tenemos a nuestra disposición en el navegador todas las carpetas que se generan y todos los componentes que contienen dichas carpetas para los procesos creados. Desarrollaremos más información de este cuadrante en el apartado de funcionalidades.
- En el tercer cuadrante, se muestran todos los componentes contenidos en el directorio del repositorio que hayamos seleccionado y abierto. En este caso en concreto de aquí podremos seleccionar los componentes que queramos y realizar las operaciones que deseemos con ellos referentes al repositorio: copiar, pegar, dar permisos, eliminar... cualquier componente elegido.
- El cuarto cuadrante es el cuadrante de la consola de logs y mensajes, en él se mostrará cuando se ha llevado a cabo una operación, la manera en la que ha finalizado ésta y sus posibles errores en caso de que los haya.

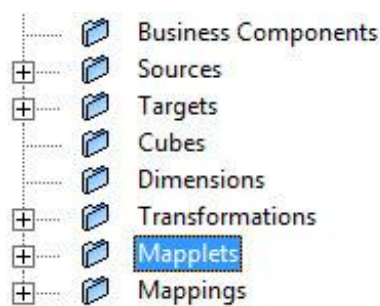
La siguiente interfaz de usuario que analizaremos será la que corresponde a la aplicación **Designer**:



**Ilustración 6: Segmentación de la interfaz del designer**

Tanto en esta aplicación como en el resto, el cuadrante 1 es igual que el ya explicado en el repository manager, varían los iconos para cada una de ellas ya que estos ofrecen funcionalidades distintas en cada aplicación, pero el aspecto es igual en todas ellas, en esta aplicación en particular y dado que en el cuadrante tres podemos establecer 5 tipos de vistas distintas para el manejo de distintos componentes del proceso, esta sección 1 variará lo que se muestra dependiendo de en que vista estemos. Las funcionalidades las explicaremos mas adelante en este mismo documento.


En el cuadrante 2, al igual que en el repository manager se muestra la vista del navegador, la diferencia con respecto a la anterior aplicación es que se muestran exclusivamente los componentes propios de esta aplicación. Las distintas carpetas que se muestran son:

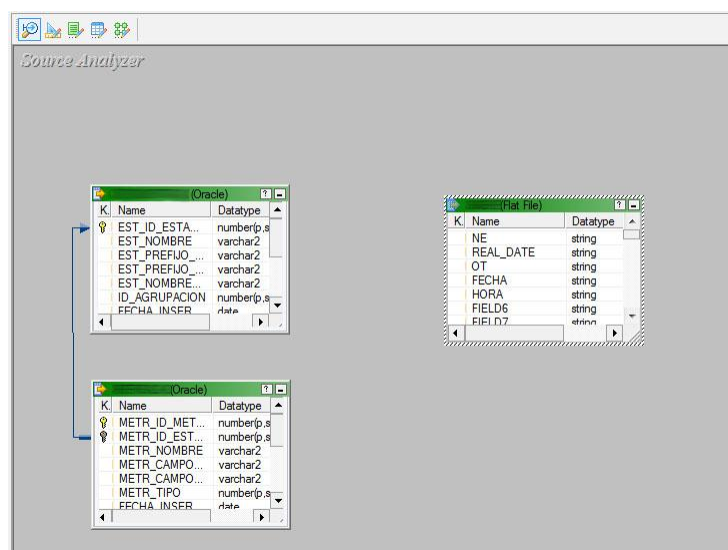


**Ilustración 7: Sistema de archivos del designer.**


- *Business components*: Son componentes diseñados para acelerar los procesos de carga de datos brutos de determinados departamentos o sectores de una empresa.
- *Sources*: Aquí se muestra el listado con todos los orígenes que se han definido para ese proceso de carga, dentro de este directorio hay a su vez subdirectorios que se crean según las fuentes que se importen y que pueden ser ficheros planos, xml o las distintas bbdd de las cuales se hayan importado esos orígenes.
- *Targets*: son destinos, tablas o ficheros en los cuales se van a insertar los datos finales.
- *Cubes*: Aquí se encontrarían los orígenes de bases de datos multidimensionales.
- *Dimensions*: Aquí se definirían las dimensiones de los cubos.
- *Transformations*: Aquí se alojan componentes creados con otras herramientas distintas a la ETL y que proporciona la misma empresa que distribuye power center. Estos componentes pueden ser muy variados.
- *Mapplets*: Son pequeños mapeos de datos que van a ser reutilizados en distintos mapeos principales y por lo cual se realiza un pequeño “programa” (o método si habláramos de un lenguaje de programación orientado a objetos) encapsulando dicha funcionalidad para que sea reutilizado tantas veces como sea necesario.
- *Mappings*: Son el componente principal que genera esta aplicación y en él se desarrolla la lógica de bajo nivel de nuestro proceso de carga.

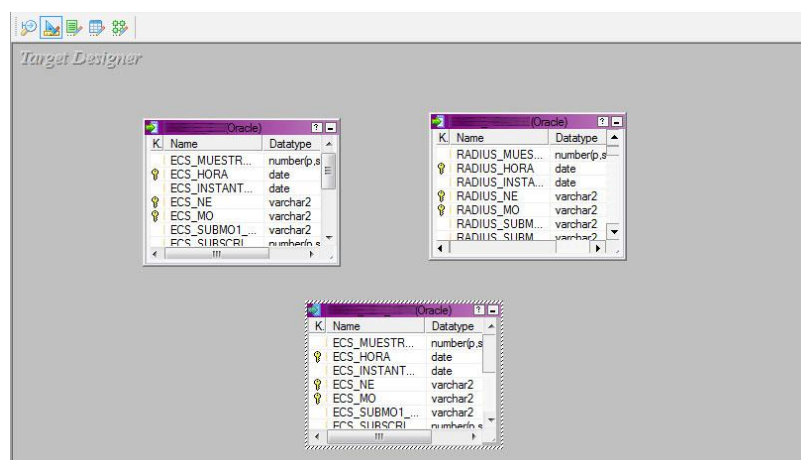
En el cuadrante 3, se muestra el elemento que hayas seleccionado y sobre el cual quieras realizar las modificaciones. Este cuadrante muestra 5 vistas que se sitúan en 5 iconos que podemos ver en la parte superior izquierda de este cuadrante (cuadro 3.1):

- **Analizador de orígenes** : En esta vista podemos importar y exportar nuestras fuentes de datos y proceder a su modificación. En esta imagen se ven dos tipos de orígenes distintos, por un lado vemos dos tablas que están relacionadas por una foreign key por el campo id\_estadistica y por otro lado vemos un origen procedente de un fichero plano.




**Ilustración 8: Analizador de orígenes.**

**Analizador de destinos** : Esta vista realiza las mismas funciones que la anteriormente comentada, con la diferencia que el producto final será la obtención de los elementos que serán contenedores de datos en lugar de fuentes de datos.



**Ilustración 9: Analizador de destinos.**

- **Desarrollador de transformaciones** : Esta vista permite crear y configurar cualquier elemento de transformación que tenga la herramienta y hacerlo reutilizable en cualquier mapping que vayamos a crear dentro de ese proyecto y que requiera esa transformación concreta.

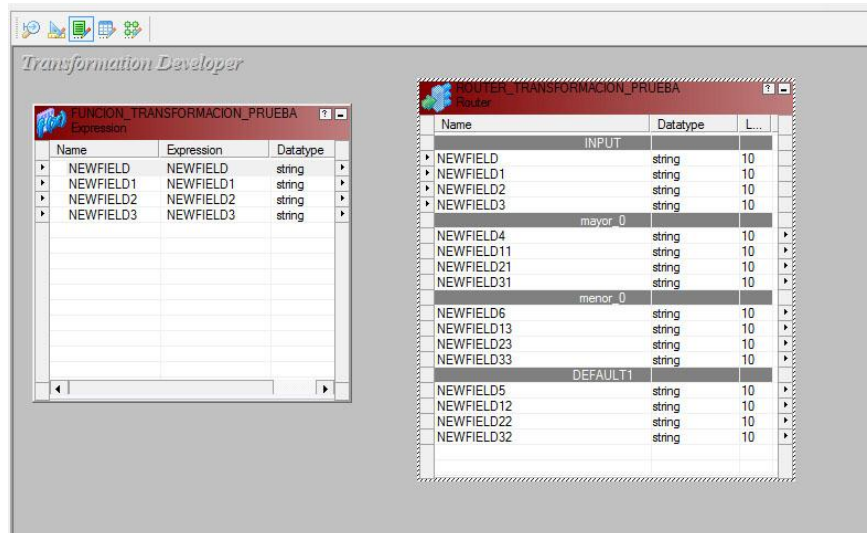



Ilustración 10: Desarrollador de transformaciones.



- Diseñador de Mapplets** : En esta vista se permite al desarrollador crear los componentes llamados mapplets, que son pequeños fragmentos de código encapsulado que se van a poder reutilizar en varios mapeos de carga. Son fragmentos de código comunes a varios procesos de carga. En la ilustración 11 se muestra la vista en la cual se realiza la lógica del mapplet, por otro lado en la ilustración 12 se ve como el código representado en la ilustración 11 se ha transformado en un componente (una caja) que podemos usar desde un mapeo y en el cual se recibirán unos campos de entrada y realizara de forma transparente la lógica implementada en la ilustración 11 y nos dará la salida correspondiente que se conectará con otros componentes para realizar la lógica final del proceso completo.

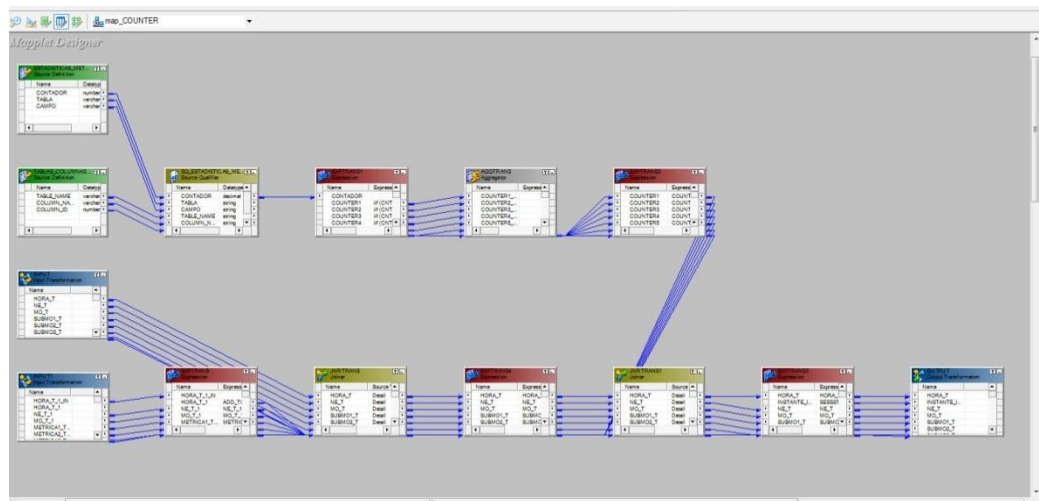


Ilustración 11: Diseñador de mapplets.

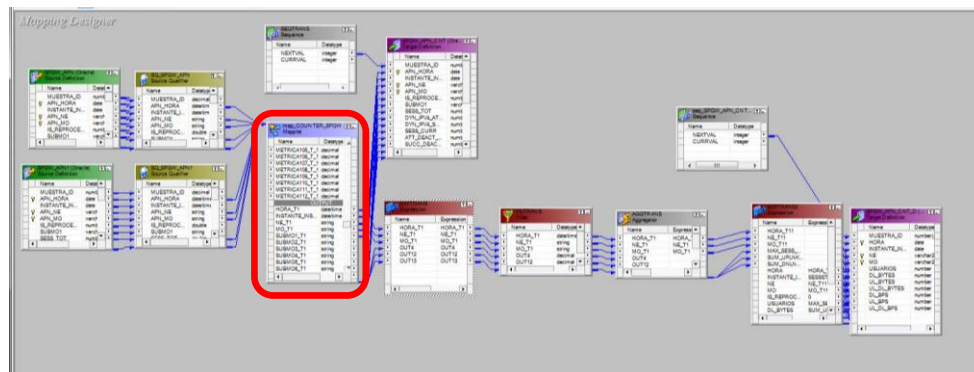



Ilustración 12: vista del uso o llamada a un mapplet.

- **Diseñador de Mappings** : En esta vista se pueden ver todos los mapeos de los procesos de carga y proceder a su modificación. En cada vista que hemos repasado podemos ver los distintos componentes en su forma “extendida” en la cual se nos muestran los campos de cada componente o por el contrario podemos verla como en la siguiente imagen en la cual se ven los iconos de cada componente sin ver cuales son sus campos, esta vista nos permite hacernos una idea de la funcionalidad del proceso sin entrar en el detalle de que campos estamos procesando.

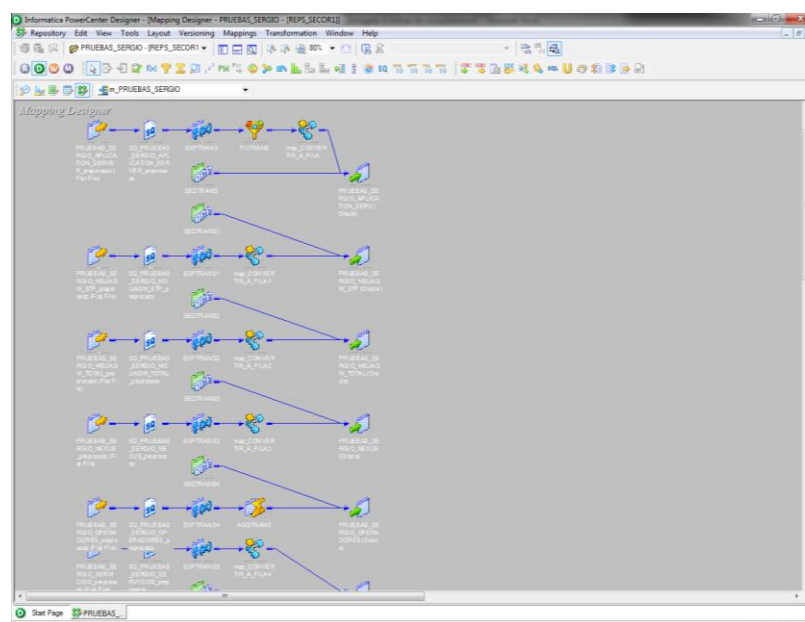


Ilustración 13: Diseñador de mappings.

Por último en el 4º cuadrante se muestran todos los mensajes que la aplicación muestra al desarrollador. La sección principal y mas usada es la ventana de salida de datos pero en el modo de depuración se pueden añadir a esta sección dos nuevas ventanas para inspeccionar en cada momento el componente que estamos tratando en ese momento y por otro lado una ventana específica para los destinos y ver que se va a insertar en cada instante.

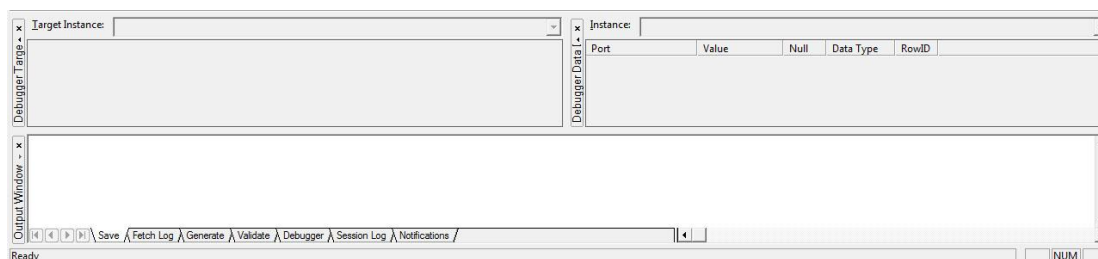
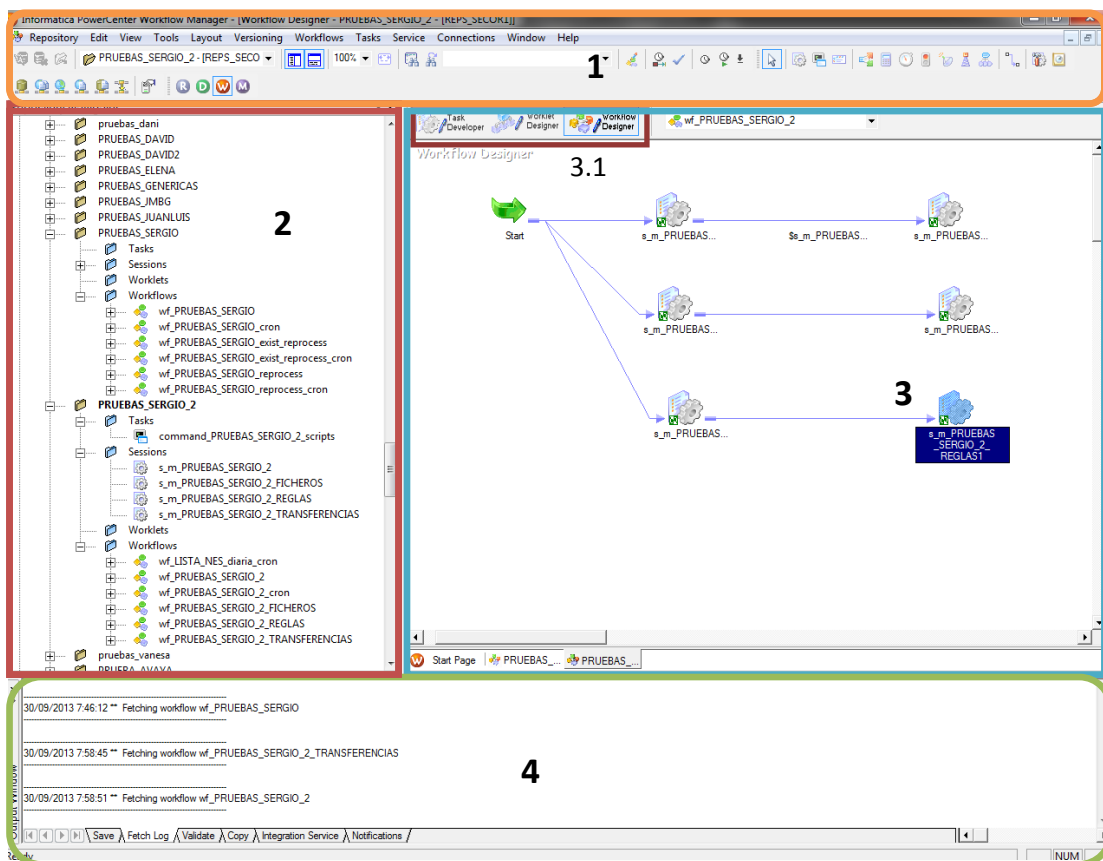


Ilustración 14: Ventanas de logs.

La ventana de salida de logs tiene 7 pestañas en las que la aplicación muestra segmentada la información oportuna para cada caso que se dé. Las pestañas son: Save, Fetch logs, generate, validate, debugger, Session log y Notifications.

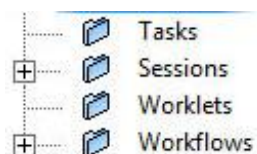
La interfaz correspondiente a la aplicación **workflow manager** es la siguiente:



**Ilustración 15: Interfaz workflow manager.**

En esta aplicación explicaremos los cuadrantes 2,3 y 4 dado que son los que difieren del resto de aplicaciones ya explicadas. El primer cuadrante es configurable y con botones propios de esta aplicación igual que en anteriores caso por lo que pasaremos a explicar en el apartado de funcionalidades.

En el segundo cuadrante, aparece como en el resto de aplicaciones el navegador de ficheros, esta aplicación tiene su propio sistema de ficheros que en este caso esta dividido en 4 directorios:



**Ilustración 16: Sistema de ficheros del workflow manager.**

- **Task:** Contiene tareas que pueden hacerse reutilizables para distintos workflows.
- **Sessions:** En este directorio se alojan las sesiones de nuestros procesos de carga, como se comentó anteriormente las sesiones son especificaciones o instancias realizadas sobre un mapping en concreto.
- **Worklets:** Son agrupaciones de sesiones de tal manera que se puede “encapsular” sesiones u operaciones relativas a trabajos con algún aspecto en común.

- **Workflows:** Son la definición concreta de la lógica de un proceso de carga al mas alto nivel. Sería como en programación orientada a objetos el hecho de crear una clase main.

En el tercer cuadrante nos encontramos similitudes con respecto al designer, aquí se muestra el contenido de los componentes que se arrastren desde los distintos directorios anteriormente indicados y dependiendo de la vista que tengamos establecida. En esta aplicación tenemos 3 tipos de vista distintas:

- En la vista denominada *Task developer* : se muestran o se pueden crear cualquier tarea de power center, estas pueden ser sesiones, comandos o componentes de envío de emails.

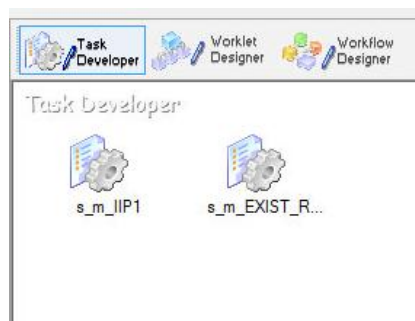


Ilustración 17: Vista sobre el Task developer.

- En la pestaña de *Worklet Designer*: Se pueden agrupar o empaquetar distintas tareas o sesiones para encapsular las funcionalidades distintas y reutilizarlas.

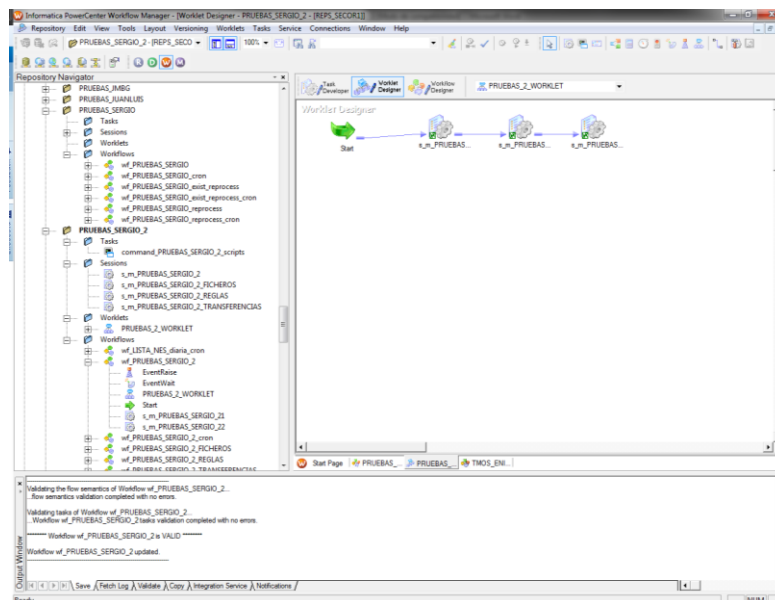


Ilustración 18: Vista sobre el worklet designer.

- Por último, la vista del *workflow designer*: En ella se crean o modifican los flujos de trabajo, los procesos de más alto nivel creando finalmente un proceso completo de extracción, transformación y carga.

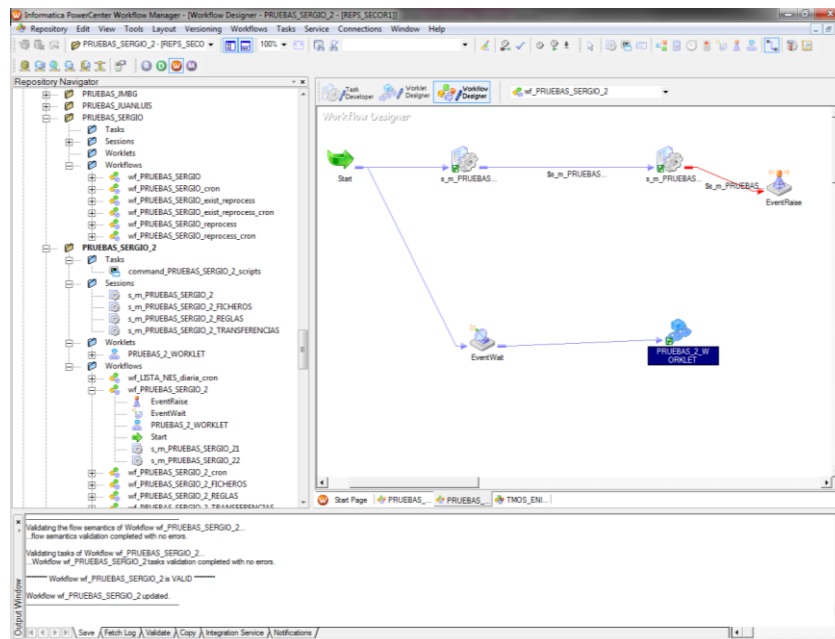


Ilustración 19: Vista sobre el workflow designer.

Finalmente en el cuadrante 4, tenemos las ventanas de logs, en este caso no se pueden añadir mas ventanas solo existe la ventana que muestra los distintos mensajes de salida divididos o filtrados en varias pestañas. Las pestañas que tenemos en esta ocasión son: save, Fetch log, validate, copy, integration service y Notifications.



Ilustración 20: Ventana de Logs del workflow manager.

La última aplicación que compone esta ETL, es la llamada *workflow monitor*, esta aplicación sirve para ver las ejecuciones de los procesos de carga y su programación, así como los logs que dejan una vez finalizados. Esta aplicación se muestra con dos vistas diferentes una en forma de diagrama de gant y otra con un listado de tareas que podemos permutar como se ve en el cuadrante 2.1.

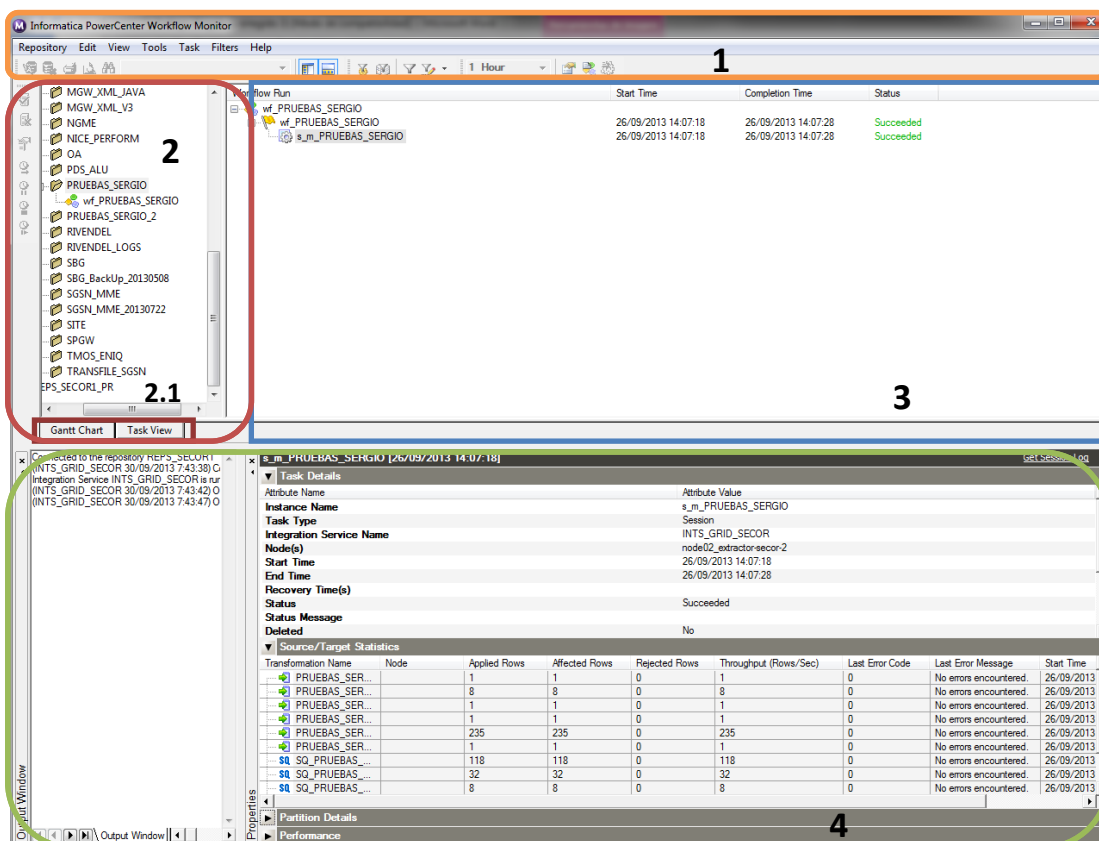
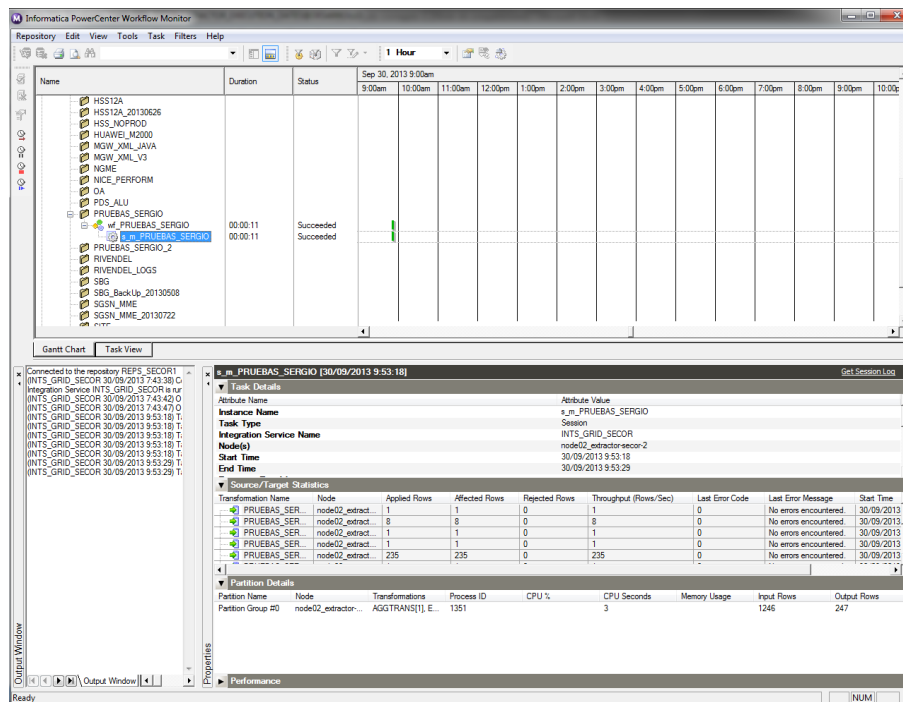


Ilustración 21: Interfaz workflow monitor, vista de tareas.

Como en las anteriores aplicaciones el cuadrante 1 tiene las barras de herramientas y los iconos de funcionalidad configurables, dichas funcionalidades se explicaran en puntos posteriores de este documento.

El cuadrante 2 muestra la estructura de directorios de procesos y nos permite seleccionar por medio de las dos pestañas señaladas en el cuadro 2.1 que vista desea el usuario que le muestre la aplicación. Las posibles vistas son en diagrama de gant sobre un eje cronológico se muestran las tareas y el tiempo consumido por el proceso u otra vista es en un listado de tareas con el estado y tiempo de proceso de las mismas tal y como se muestra en la ilustración 20.

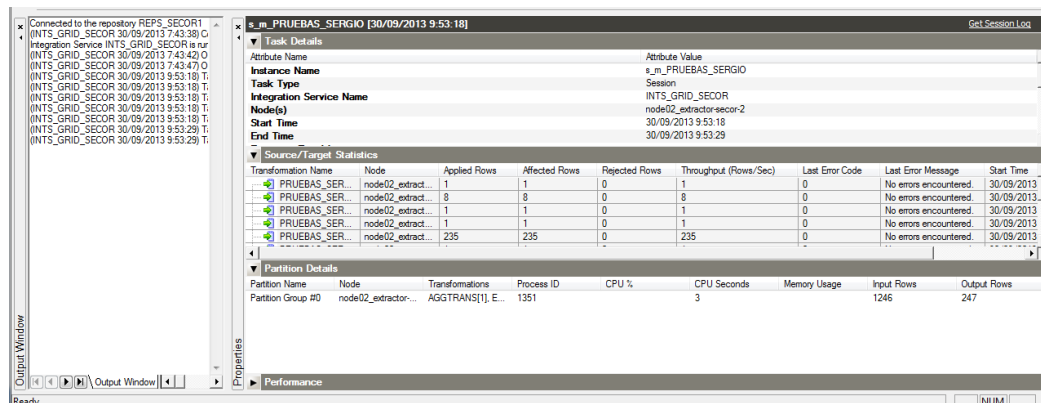
En el cuadrante 3 se muestra toda la información de los extractores seleccionados en el menú de navegación y además se plasma la vista seleccionada en el apartado 2.1. Desde aquí podremos ver si las tareas se han ejecutado bien o mal y al seleccionarlasy veremos sus logs en el apartado 4. Aquí podemos ver la ejecución de los workflows que son los elementos simbolizados por una bandera amarilla y los componentes o tareas de los mismos que son lo denominado sesiones, que componen cada workflow y que se simbolizan por una rueda de engranaje.



**Ilustración 22: Vista de las ejecuciones en diagrama de gant.**

El 4º cuadrante tiene las vistas de logs de las tareas ejecutadas, tiene 5 ventanas, la de la izquierda muestra las tareas ejecutadas en el servidor de integración y las 4 de la derecha muestran los logs. Estas últimas mostrarán información distinta según seleccionemos un workflow (el del icono de la bandera) o una sesión (la del icono del engranaje).

Cuando seleccionamos una sesión se muestran los siguientes datos:



**Ilustración 23: vista de la ejecución de una sesión.**

Se muestran cuatro ventanas en las que nos dan la siguiente información:

- En task Details: nos indica todo lo referente a la maquina donde se esta ejecutando y los detalles del proceso ejecutado.
- En la segunda ventana vemos los datos concernientes a las lecturas de orígenes y las escrituras en los destinos, maquinas donde se ejecutan, tasa de lectura y escritura, numero de registros leídos o escritos, tiempo de duración, etc.

- En la tercera ventana se muestra, si activamos un modulo de particionado, en que maquinas se esta ejecutando una sesión o un extractor de manera que se pueda ejecutar en paralelo en varias maquinas.
- En la cuarta ventana llamada performance, se muestran datos cuando en la configuración de la sesión así se lo indicamos, al estar activada dicha opción podemos ver en esta ventana que elementos de una sesión se están ejecutando y distintos datos como tiempo que tarda en cada elemento, uso de la maquina en ese momento...

Por otro lado si se selecciona un workflow los datos que se muestran son los siguientes:

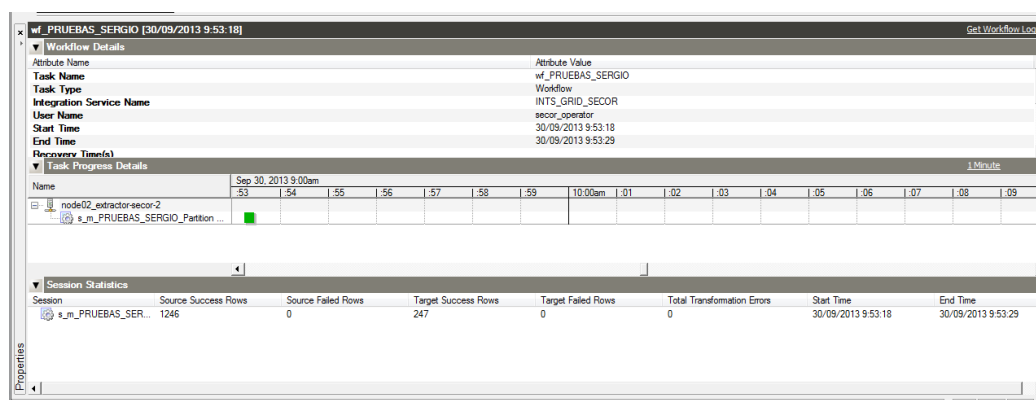


Ilustración 24: vista de la ejecución de un Workflow.

En esta imagen podemos ver como la primera ventana muestra los mismos datos que la anterior de las sesiones pero ahora con datos correspondientes al workflow.

En la segunda ventana se muestra un diagrama de gant con las ejecuciones de todas las sesiones de un workflow y muestra en que maquinas se esta ejecutando (ya que en esta arquitectura hay dos maquinas una espejo de otra que permite ejecuciones en paralelo y balancea la ejecución de sesiones según la carga de las dos maquinas) y la duración de estas ejecuciones.

La última sesión muestra las estadísticas de las sesiones del workflow ejecutados: registros leídos y escritos, registros rechazados tiempos de inicio y fin y más detalles que se pueden configurar para que se muestren en esta vista.





### 2.6.1.1 Conexión y desconexión a repositorios.

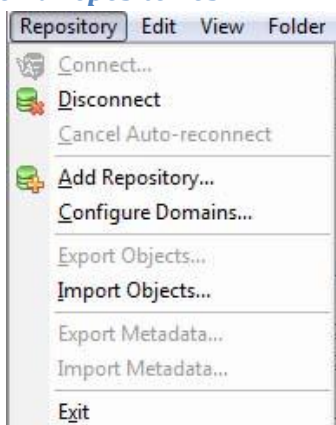


Ilustración 26: Menú repository.

Aquí se ve como se puede conectar o desconectar de los repositorios configurados en la ETL. Si le damos a conectar nos saldrá una ventana de login para que introduzcamos un usuario y una contraseña e indiquemos a que repositorio queremos conectarnos. Esta misma funcionalidad la podemos obtener realizando doble clic sobre cualquiera de los repositorios configurados en la aplicación en el apartado de repository navigator que pudimos ver en el apartado de interfaz en la [ilustración 5](#), cuadrante 2, en el cual nos aparece un icono verde como si fuera una bbdd o una pila de círculos. También se puede configurar que se cancele la conexión automática si por cualquier motivo se ha perdido la conexión con el repositorio.

### 2.6.1.2 Configuración de dominios y repositorios.

Otra funcionalidad que nos ofrece este menú es la de añadir repositorios y configurar dominios. Los dominios son como “contenedores” de repositorios y los repositorios son como sistemas de ficheros que contienen los distintos componentes de los procesos de carga de la ETL. Con el configure domains, podremos especificar en que maquina o maquinas van a estar contenidos el o los repositorios y que puerto usamos para conectarnos.

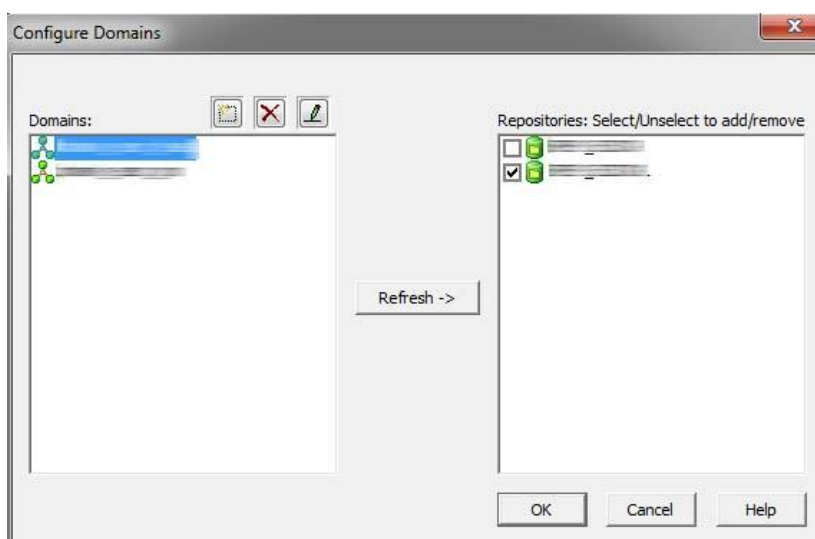


Ilustración 27: configuración de dominios.

Aquí podremos crear, eliminar o modificar nuevos dominios sobre la maquina en la que esta alojado power center, realmente desde la parte de cliente solo haces referencia o configuras dominios que ya están creados en la parte servidor (el alcance de este proyecto no llega hasta la parte servidora de la aplicación). En esta ilustración podemos ver a la izquierda dos dominios y a la derecha como sobre uno de los dominios que hemos seleccionado hay configurados dos repositorios aunque solo vamos a utilizar uno de ellos que está marcado con el tic.

Si le damos a editar uno de los dominios veremos como para ese nombre de dominio nos deja configurar el host y el puerto a través del cual se va a conectar el repositorio.

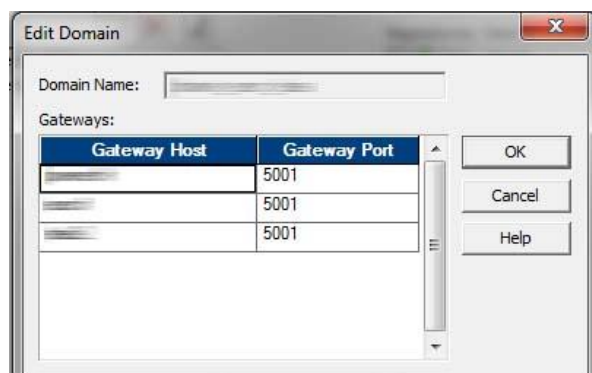


Ilustración 28: Edición de dominios.

### 2.6.1.3 Exportar e importar objetos:

Por otro lado desde este menú se nos permiten dos opciones de importar y exportar, por un lado solo la parte de estructura, importando o exportando metadatos y por otro lado la opción de importar o exportar los objetos con todas sus configuraciones y dependencias.

La funcionalidad mas usada es la de importar/exportar un objeto, para ello seleccionamos el componente que se desea exportar y pulsamos en la opción “Export objets” de la barra de menú Repository, esto nos abrirá una ventana de explorador en la que nos indicará donde se desea guardar el fichero .xml que genera con todos los datos de los objetos del componente exportado.

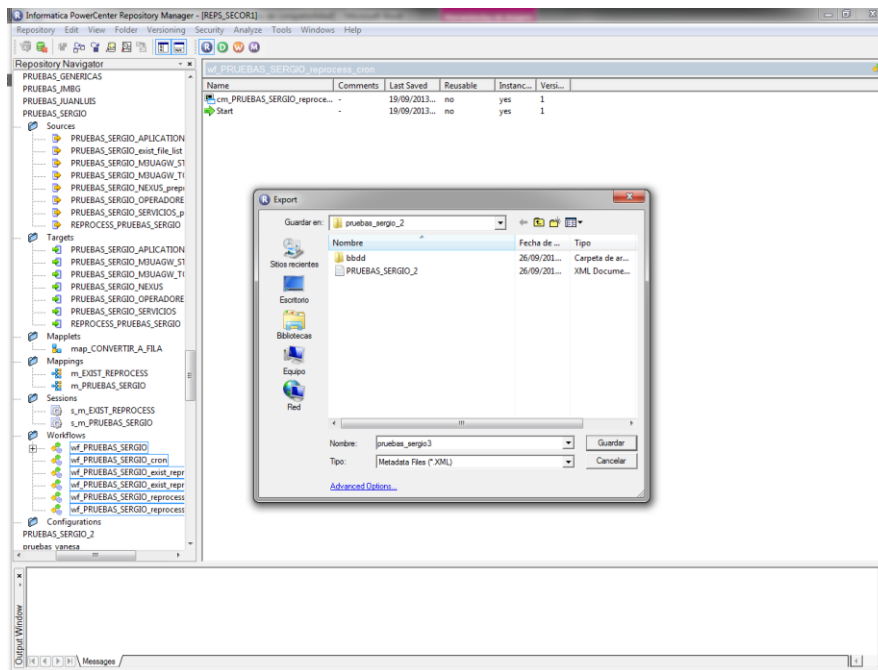


Ilustración 29: Exportar objeto paso 1.

Una vez finalizada la exportación del componente nos sale un cuadro de dialogo con logs de salida acerca de que objetos pertenecientes a ese componente se han exportado y si ha habido algún error o warning.

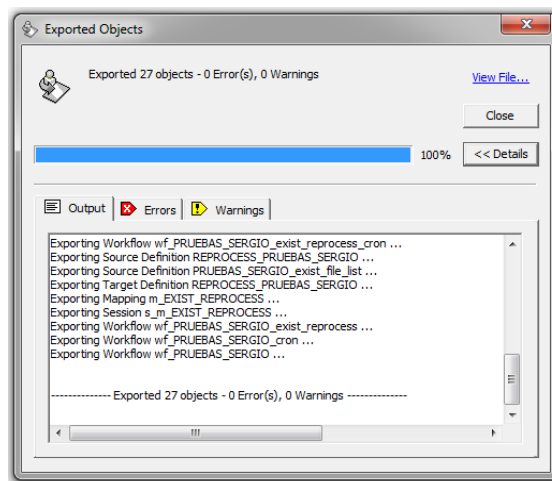


Ilustración 30: Exportar objeto, paso 2.

Para importar un componente que hayamos exportado previamente deberemos ir al menú repository, seleccionar “import objects” de tal manera que se abrirá una ventana de asistente de configuración que nos guiará paso a paso acerca de como realizar la importación.

En primer lugar nos indicaran que seleccionemos el fichero .xml que anteriormente hayamos exportado, para ello tendremos que buscar e indicar la ruta completa hasta el fichero y clicar en siguiente.

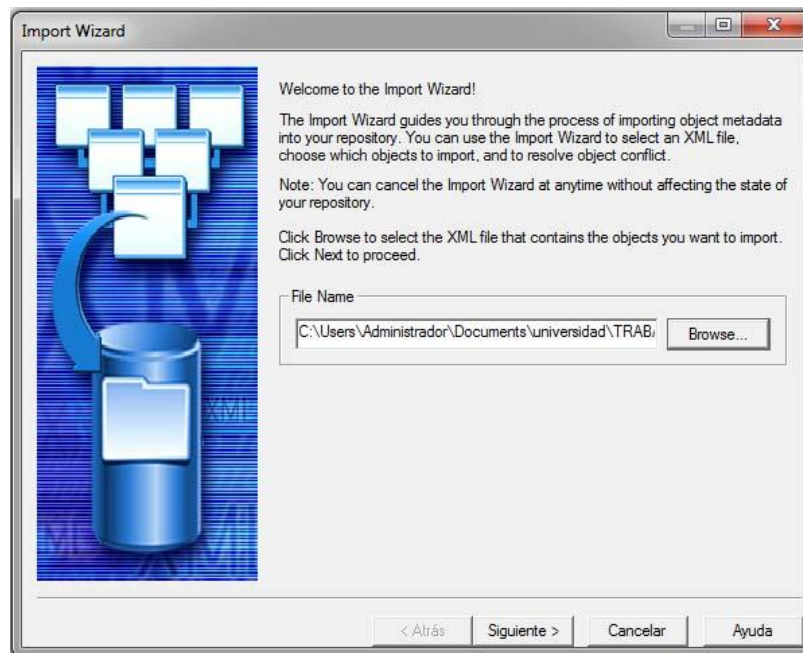


Ilustración 31: Asistente de ayuda import, paso 1.

La siguiente ventana, nos mostrará dos cuadros uno, a la izquierda, con la estructura listada de objetos que tenía el componente anteriormente importado y otro, a la derecha, con los objetos que deseamos importar de ese componente. Al exportar un componente si este tuviera objetos que son referencias de otro componente, también se importan los objetos del componente que los contiene. En este ejemplo que se muestra en la ilustración 31 vemos como en los targets (destinos) sus nombres comienzan con la palabra *shortcut* esto significa que estos *shortcut* son como “punteros” a otro componente que es el que tiene realmente dichos objetos de destino, esto se hace porque cuando dos componentes comparten algún objeto si este objeto es modificado y se modifica en el componente que tiene el origen, todos los componentes que tengan las referencias (shortcuts) a esos objetos se modificaran automáticamente.

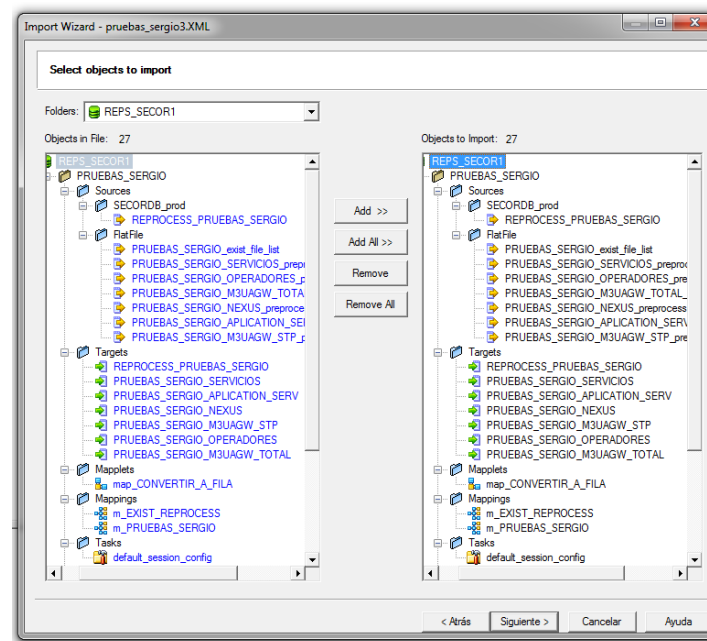
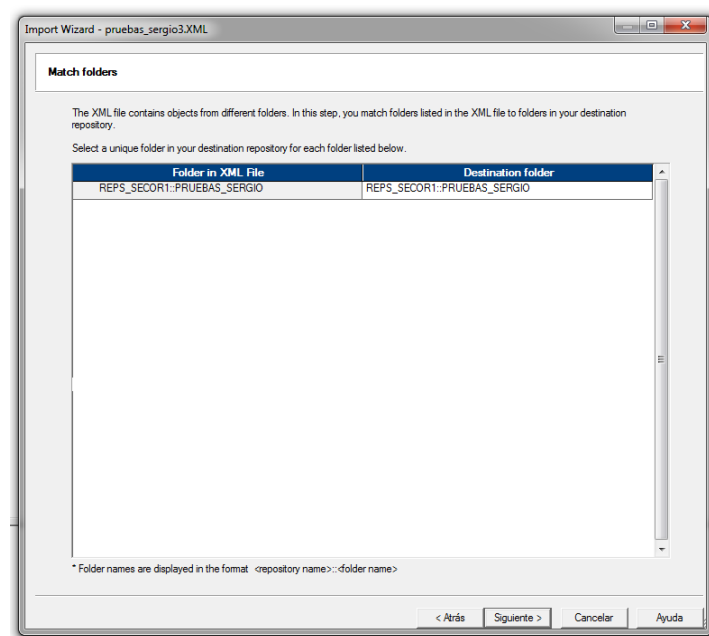


Ilustración 32: Asistente de ayuda import, paso 2.

En la siguiente ventana se muestra en la parte izquierda como se llamaban los directorios donde estaban alojados esos componentes y nos solicita en la derecha que indiquemos si queremos que se guarde en un directorio con el mismo nombre o si queremos que se cree un nuevo directorio con otro nombre distinto.



**Ilustración 33: asistente de ayuda import, paso 3.**

La siguiente ventana mostrara un cuadro para crear posibles reglas en caso de que existan conflictos entre objetos, normalmente no se crean reglas y si existiera algún conflicto cuando le das a siguiente en esta ventana, saldrá una nueva ventana con un listado de conflictos y una serie de opciones para su resolución. En nuestro caso particular nos aparecen tres conflictos con un workflow, una sesión y un mapping. El conflicto que ocurre es que estamos intentando importar sobre un directorio que ya existe y por lo tanto estos objetos existen en ese directorio (esto es porque estoy realizando el export y el import sobre el mismo repositorio, he realizado a propósito esta prueba para que surgieran estos conflictos y poder detallarlos), de tal manera que me ofrece 4 posibles soluciones para resolver el problema: renombrar el objeto en conflicto, remplazar el existente por el nuevo, reutilizar el que ya existía puesto que no hay diferencias o ignorar este conflicto, señalaremos la opción que más nos convenga y le daremos a siguiente. Al señalar una de las opciones nos saldrá si deseamos aplicar esta misma solución para todos los conflictos que nos han listado. Una vez que hemos indicado que deseamos hacer con dichos conflictos y dado a siguiente aparece una ventana resumen de los conflictos tratados y la solución optada, en la cual aparecerá un botón de cerrar dicha ventana y los conflictos que darán solucionados. Mostrando de nuevo la ventana anteriormente mencionada para la creación de reglas de resolución de conflictos pero esta vez con los conflictos ya solucionados.

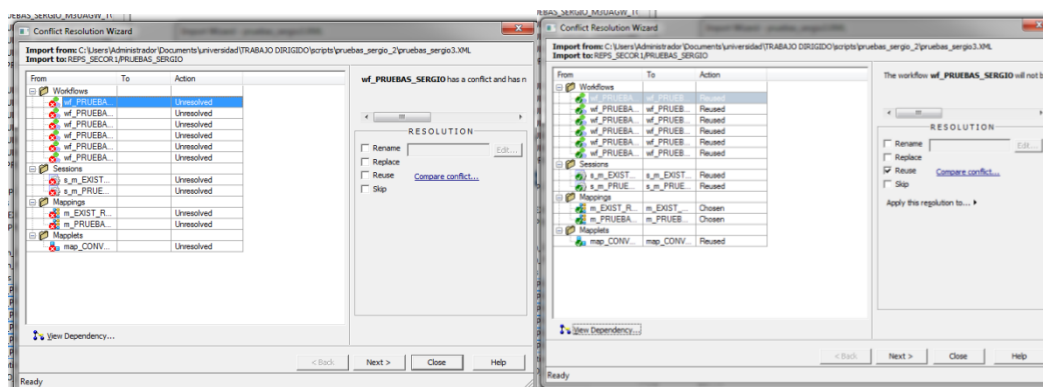


Ilustración 34: Asistente ayuda solución de conflictos import.

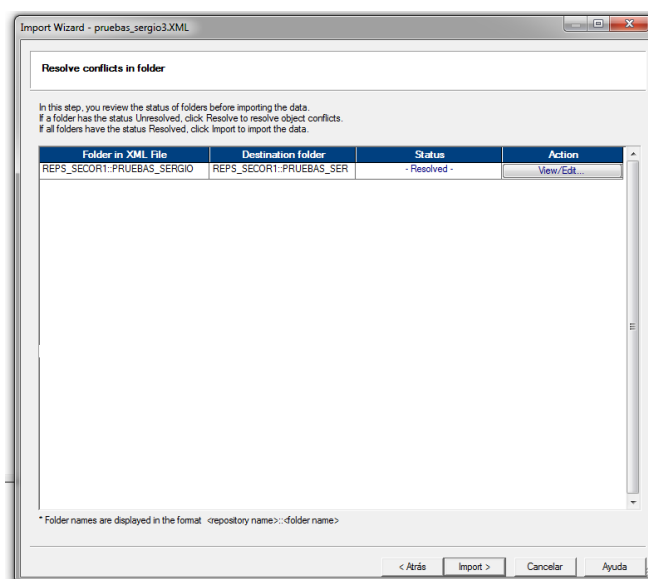


Ilustración 35: Asistente de ayuda, resumen de resolución de conflictos.

Una vez llegado a este paso le daremos al botón de importar y empezará a generar el componente y todos sus objetos. Una vez finalizado el proceso saldrá una ventana con los logs del proceso de importación y una vez le demos al botón done, se cerrará la venta y habrá concluido el proceso definitivamente.



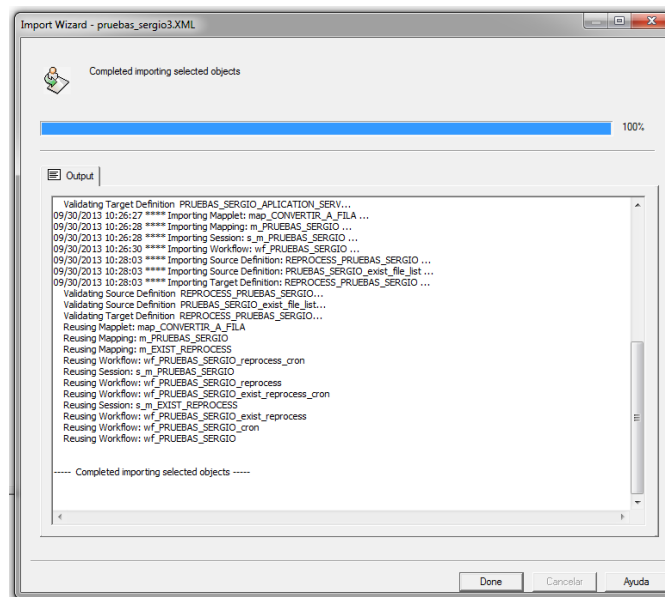


Ilustración 36: Proceso de importación.

#### 2.6.1.4 Crear-borrar-editar directorios en el repositorio:

Desde el menú “folder” de la barra de menús se pueden crear, editar o borrar directorios en el repositorio.

Para crear un directorio pincharemos en el menú indicado y daremos a “create”.

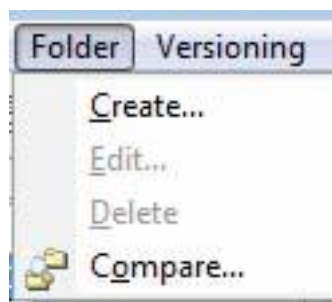
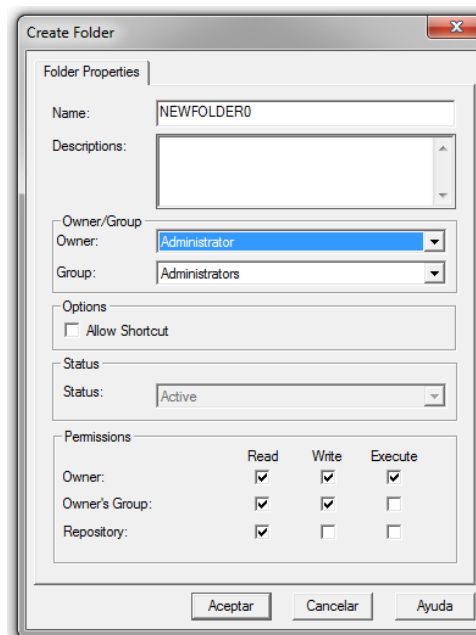


Ilustración 37: Menú Folder.

Seguidamente aparecerá una ventana en la cual podremos crear y configurar los atributos del directorio que se va a crear. En ella se podrá indicar el nombre del directorio, el propietario de la carpeta y el grupo al que pertenece el propietario de dicha carpeta y los permisos que deseamos darle a ella. También se puede indicar si permitiremos que en ella se almacenen fuentes que se podrán referenciar desde otros directorios como shortcuts, esta será una carpeta compartida. Por defecto al crear un directorio el estado del directorio nos aparece como activo y no podemos cambiarlo.



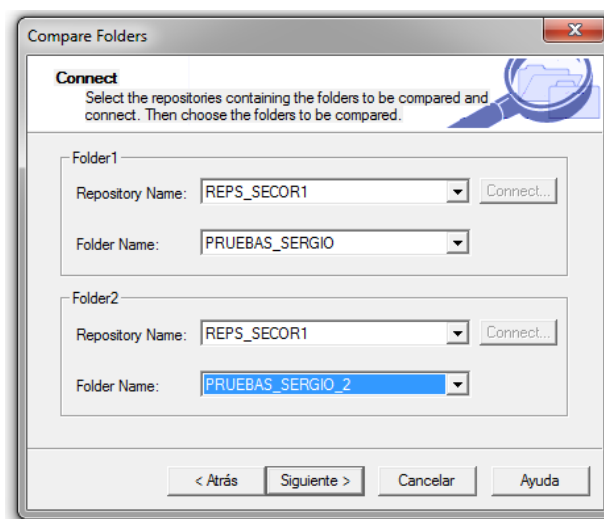
**Ilustración 38: Creación de un directorio en el repositorio.**

Al editar un directorio aparece la misma ventana que se muestra en la anterior ilustración y nos permite modificar los campos para y permisos que requiramos modificar.

El borrado de ficheros se realiza marcando en el navegador de directorios el que quieres borrar y dándole en el menú folder a la opción de borrar. Nos solicitara confirmación de la operación, una vez confirmado se procede al borrado del directorio

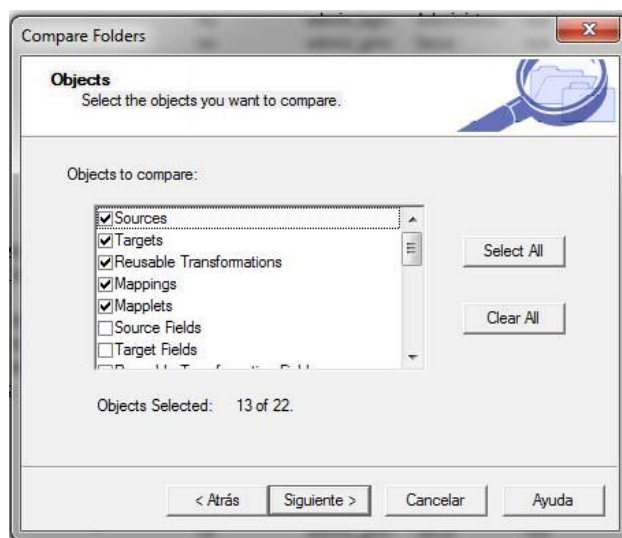
#### **2.6.1.5 Comparador de directorios en el repositorio:**

Otra funcionalidad que ofrece esta aplicación es la de poder comparar el contenido de dos directorios que haya en el repositorio. Para ello debemos ir al menú “Folder/compare Folders”. Cuando seleccione esta opción saldrá un asistente que nos guiará en todo el proceso.



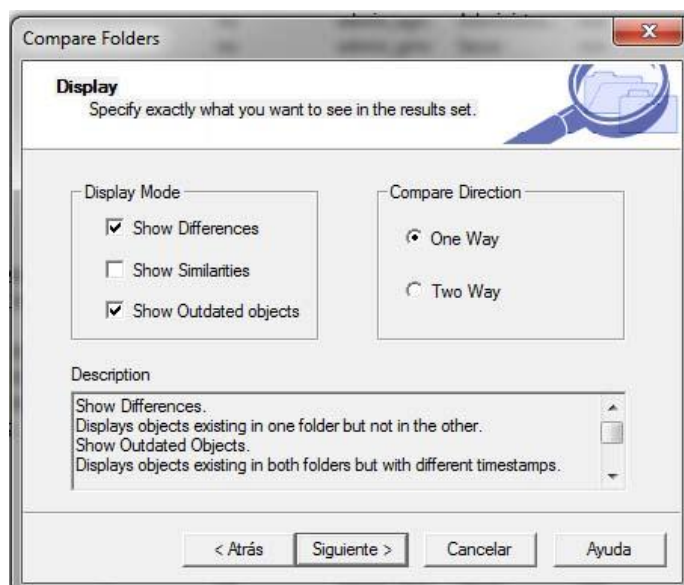
**Ilustración 39: comparador de directorios, paso 1.**

En primer lugar se tiene que seleccionar que directorios y en que repositorios están cada uno de dichos directorios. Se pueden comparar los contenidos de directorios de distintos repositorios.



**Ilustración 40: Comparador de directorios, paso 2.**

En el segundo paso, se tiene que indicar que objetos contenidos en esos directorios serán los que se desean comparar, seleccionándolos marcando el check box.



**Ilustración 41: Comparador de directorios, paso 3.**

En el tercer paso, hay que seleccionar el que deseamos que muestre el informe de salida del comparador, podemos elegir entre que muestre las diferencias entre ambos directorios y objetos seleccionados (aquí el objeto existe en ambos directorios pero hay cambios entre ellos), que se muestren las similitudes (en este caso se muestran los objetos que sean exactamente iguales en un directorio que en otro y que no hayan sufrido cambios) y por último los objetos que están obsoletos (que son aquellos que ya no tienen una relación entre un directorio y otro, es decir, que ya no existe en alguno de los dos directorios). Se puede indicar también si queremos que la comparación sea unidireccional, es decir, que coja un

directorio como maestro y compare con el segundo seleccionado o si por el contrario queremos que sea bidireccional, en la cual se compararán ambos directorios usando ambos como maestros.

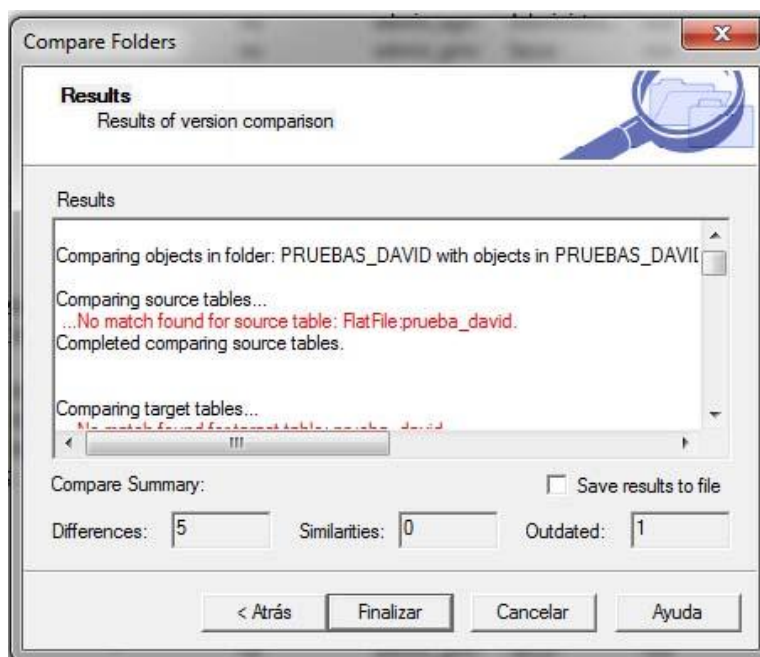


Ilustración 42: Comparador de directorios, paso 4.

Por último, se mostrará un informe en una ventana con los resultados de la comparación, dicho informe se podrá salvar a un fichero, si marcamos el check que indica “save results to file”.

En la ventana de resultados a parte del detalle de la comparación se muestra un resumen con los resultados obtenidos, en él se indican los objetos que ha encontrado diferentes, los que son iguales y los objetos que ya han sido borrados en alguna de las carpetas comparadas.

Tras pulsar en finalizar desaparecerá esta ventana y si le marcamos la opción de salvar a un fichero se habrá creado dicho fichero en la ruta indicada con los resultados obtenidos.

#### 2.6.1.6 Funciones de seguridad:

En el menú “Security” se tienen dos opciones posibles: la primera es la de cambiar la contraseña del usuario con el que estas registrado y la segunda es la gestión de los usuarios, grupos y permisos de los mismos (esto siempre que seas administrador de la aplicación, aquí también podrás resetear la password de otros usuarios).

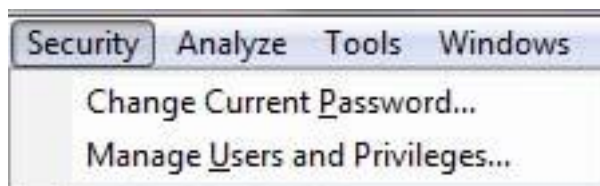
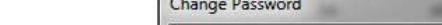


Ilustración 43: Menú Security.

Para cambiar la contraseña vigente del usuario que esta registrado iremos a la opción del menú “Change current password”, una vez pulsada, saldrá una ventana en la cual saldrá



Change Password

User Name:

Old Password:

New Password:






















Retype New Password:

OK Cancel Help

La segunda opción que se ofrece en este apartado es toda la referente a la gestión de usuarios y sus permisos.

Manage Users and Privileges

Users | Groups | Privileges

Name ▲	Status	Group Names	Description
 Enabled	Enabled	Administrato...	
 Enabled	Enabled	Administrato...	
 Enabled	Enabled	Administrators	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Secor, Dev...	
 Enabled	Enabled	Public	
 Enabled	Enabled	Public	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Public	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Administrators	
 Enabled	Enabled	Administrators	
 Enabled	Enabled	Default	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Public	
 Enabled	Enabled	Developers,...	
 Enabled	Enabled	Operators, S...	

Add...  
Remove  
Edit...  
Enable  
Disable

Aceptar Cancelar Aplicar Ayuda

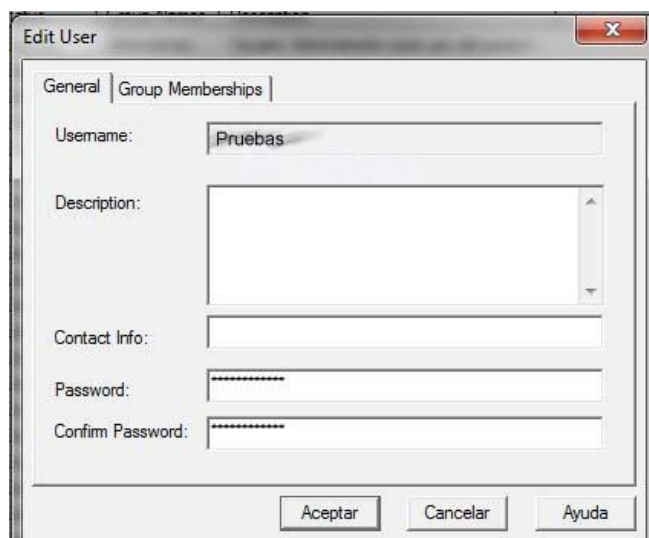
Aquí se muestra el listado con todos los usuarios creados para la aplicación, además se ve si están activos o inactivos, el grupo al que pertenece dicho usuario (esto le dará unos privilegios u otros) y una breve descripción de quién es ese usuario y para que se usa.

En esta ventana podremos además crear, borrar, editar y habilitar o deshabilitar usuarios.

La opción de borrar usuarios se aplica seleccionando un usuario y una vez que esté marcado dicho usuario se pulsara en el botón “*remove*”, saldrá un pop up solicitando confirmación acerca de la operación indicada y pulsando en aceptar el usuario quedará borrado.

Pulsando en el botón “*Enable/Disable*”, una vez que hallamos seleccionado un usuario se cambiará el status de dicho usuario, Una vez deshabilitado un usuario no podrá usarse, hasta que se vuelva a cambiar su estado a habilitado.

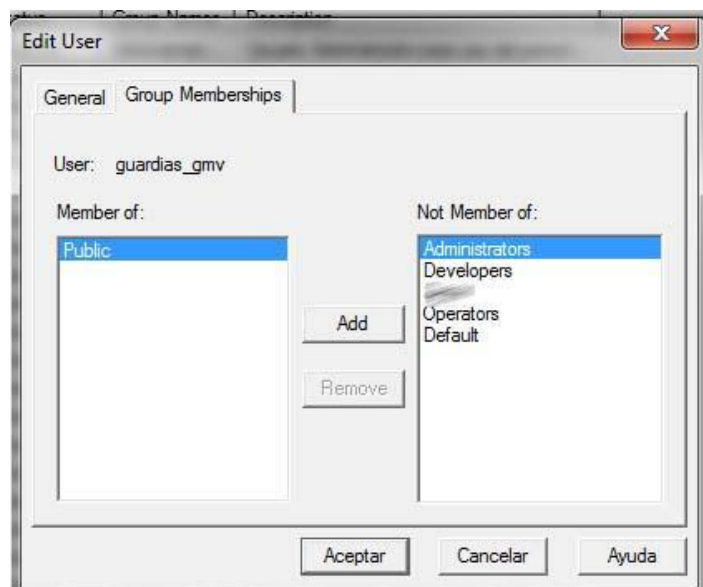
Para crear un usuario, pulsaremos en el botón que aparece a la derecha en el cual pone “*add*” y se mostrara la ventana siguiente.

The image shows a Windows-style dialog box titled "Edit User". It has two tabs: "General" and "Group Memberships". The "General" tab is active. Inside the dialog, there are several input fields: "Username" with the text "Pruebas", "Description" with a large empty text area, "Contact Info" with a single-line text field, "Password" with a masked field (asterisks), and "Confirm Password" with a masked field (asterisks). At the bottom right, there are three buttons: "Aceptar", "Cancelar", and "Ayuda".

**Ilustración 46: Crear o editar usuario.**

Aquí podremos definir un nuevo usuario y crear una contraseña para dicho usuario. Esta misma venta se muestra en la edición de usuarios.

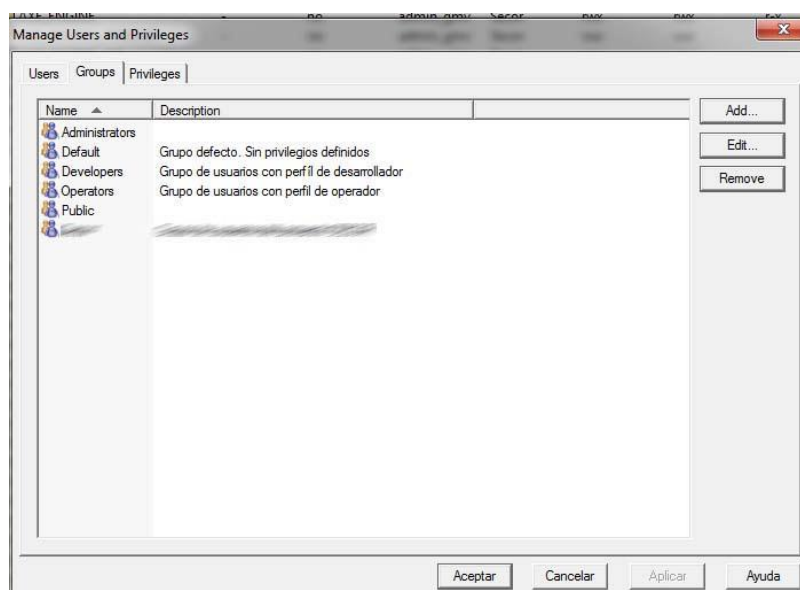
Una vez rellenos los campos que se desean informar acerca del nuevo usuario se le puede asignar a uno o varios grupos de usuarios, estos grupos tendrán definidos los privilegios que adquirirá este usuario de cara al uso de la aplicación. Lo normal y aconsejable es que se aplique únicamente un solo grupo.



**Ilustración 47: Asignar un usuario a un grupo.**

En esta ventana vemos todos los grupos de usuarios y seleccionando y pulsando o en “add” o “remove” iremos asignando dicho usuario a los distintos grupos. El cuadro de la izquierda contiene los grupos a los cuales pertenece el usuario y el cuadro de la derecha a los cuales no pertenece.

Desde la ventana principal pulsando en la segunda pestaña aparecerán las opciones sobre los grupos de usuarios. En primer lugar se mostrarán todos los grupos creados con una pequeña descripción de qué es cada grupo y en la parte derecha de la ventana aparecerán las opciones de administración sobre estos grupos. Se podrá crear y editar grupos (tienen la misma interfaz, lo único que en la creación los campos de inicio te vienen vacíos y en la edición vienen con los campos rellenos con los valores actuales) y borrar grupos.



**Ilustración 48: Administrar grupos de usuarios.**



Para borrar un grupo habrá que seleccionar el grupo que se desee borrar y pulsar en el botón “remove”, aparecerá una ventana de pop up que nos solicitará confirmación sobre la operación elegida y dándole a aceptar, el grupo quedará eliminado.

Para crear un grupo se pulsará sobre la opción “add”, entonces se mostrará la siguiente ventana. En ella podremos introducir o modificar el nombre del grupo que se desea crear, la descripción acerca del grupo y alguna información de contacto.

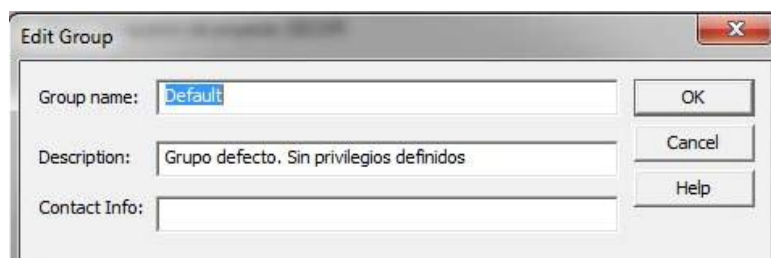


Ilustración 49: Crear/editar grupo.

Por último, la ventana principal tiene una tercera pestaña en la cual se relacionan los usuarios y los grupos con los permisos que se desean dar al usuario o al grupo creado para las distintas aplicaciones. Si un usuario pertenece a un grupo y este grupo tiene definidos unos permisos, ese usuario tendrá dichos permisos por pertenecer a ese grupo, pero además puede existir la posibilidad de que un usuario no pertenezca a un grupo, en cuyo caso se le asignarán permisos propios para ese usuario o que aun siendo un usuario perteneciente a un grupo, se le quieran dar más permisos de los que tiene su grupo en concreto.

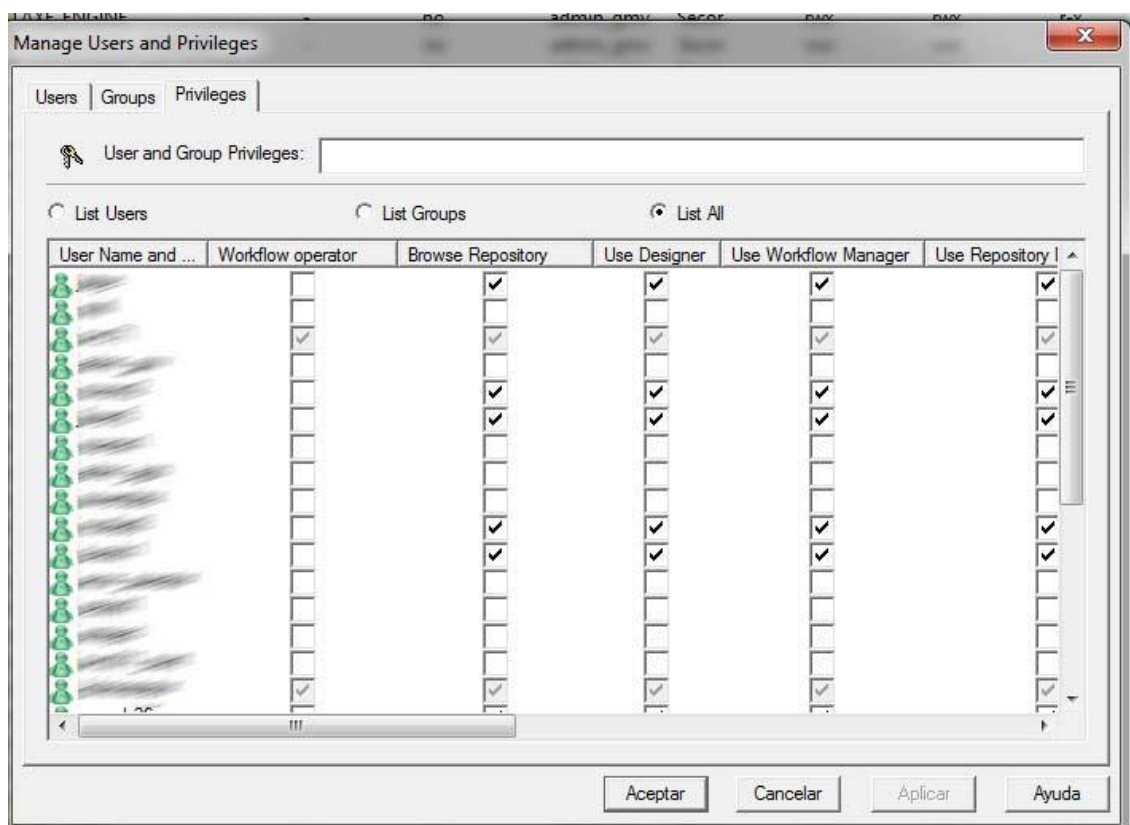


Ilustración 50: Listado de privilegios para usuarios y grupos.



### 2.6.1.7 Otras funcionalidades de la aplicación:

A parte de las ya citadas, hay otras posibles funcionalidades que ofrece la aplicación pero que en nuestro caso no aplican ya que nuestra licencia no incluye el paquete para esa funcionalidad o bien porque no tengamos instalado el sistema que se integra con dicha funcionalidad.

Para ilustrar este ejemplo, podemos ver como tenemos en la barra de menú la opción de tener un gestor de versiones que nos permita tener mas controlado el contenido de nuestro repositorio, creando distintas versiones para los desarrollos y que nos facilite más su gestión y marcha atrás. Esta opción se podrá usar una vez tengamos un gestor de versiones instalado y vinculado a la aplicación, cuando este activa nos permitirá realizar como se muestra en la imagen todas las opciones propias que ofrece una aplicación de este tipo.

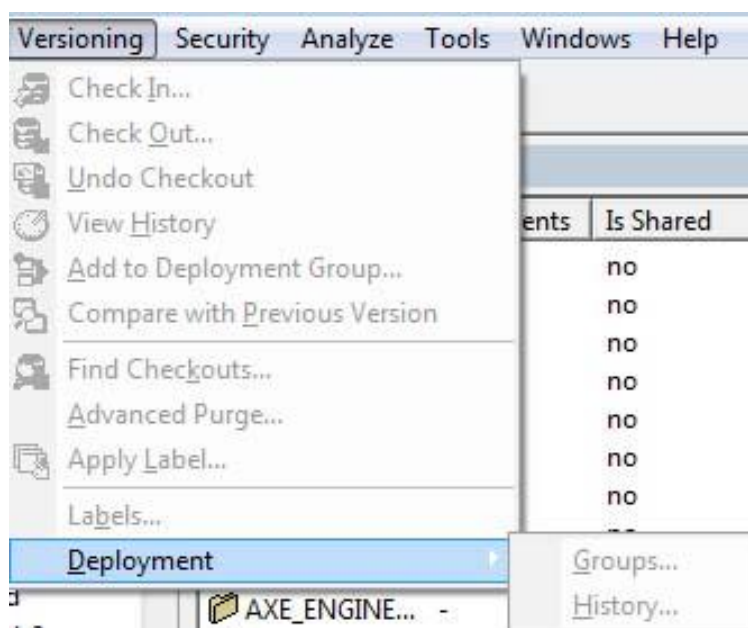


Ilustración 51: Menú de versionado del repositorio.

En el menú "Analyze", se ofrecen las opciones de analizar las distintas relaciones que tiene un objeto seleccionado en el repository manager.

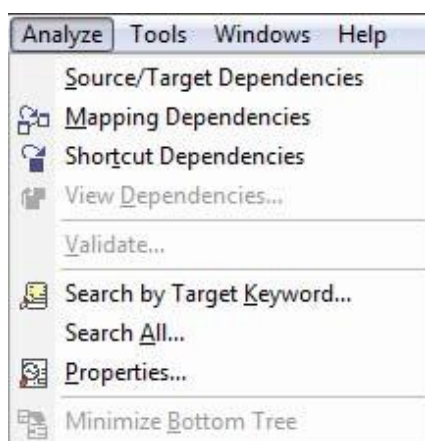


Ilustración 52: Menú de análisis.

La opción “*Source/Target dependencies*”, mostrará al desarrollador los objetos de tipo fuente o destino que tiene el objeto seleccionado. Dicho de otra manera, para un objeto seleccionado de cualquier tipo: mapping, workflow, sesión, etc, nos dirá todos los orígenes y todos los destinos que tiene asociados dicho objeto. Lo mismo ocurre con las opciones “*Mapping dependencies*”, aunque en este caso se indicaran todos los objetos que están relacionados con un mapeo seleccionado y “*Shortcut Dependencies*” esta funcionalidad nos indica todas las relaciones establecidas entre un shortcut y el resto de objetos que lo utilicen.

El menú “*Tools*” de la herramienta presenta las siguientes opciones:

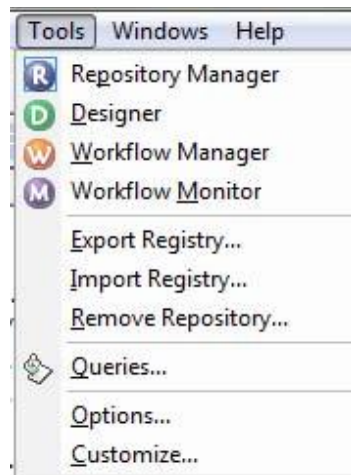


Ilustración 53: Menú de utilidades.

En primer lugar ofrece accesos directos al resto de aplicaciones de la herramienta. Si un usuario se encuentra logado en una herramienta y pulsa cualquier acceso directo a cualquier otra de las aplicaciones, esta aplicación se abrirá y logará automáticamente, además abrirá el directorio o directorios sobre los cuales estuviera trabajando el desarrollador.

Por otro lado permite las funcionalidades de importar o exportar objetos a ficheros .xml y eliminar repositorios ya creados.

La opción de Queries, permite realizar una búsqueda sobre la aplicación de cualquier elemento que esta contenga y aplicarle filtros a esa búsqueda. Esto es posible dado que la aplicación tiene por detrás una base de datos en la cual se tiene inventariado todo el contenido que guarda el repositorio.

## 2.6.2 Funcionalidades del mapping designer.

Esta aplicación contiene la mayor parte de la funcionalidad que ofrece la herramienta y es la base de todos los procesos que hagamos sobre la bbdd. En ella se diseñarán los mapeos que serán la lógica del proceso en su más bajo nivel.

Las funcionalidades que se podrán utilizar serán accesibles desde la barra de herramientas y desde los menús de los que dispone la aplicación, estos serán ligeramente distintos según la vista en la que nos encontremos de las 5 vistas posibles que ofrece la aplicación, estas diferentes vistas se explicaron en el apartado “3.2 Interfaz” del presente documento. La barra de herramientas que se muestra al iniciar la aplicación es la siguiente:

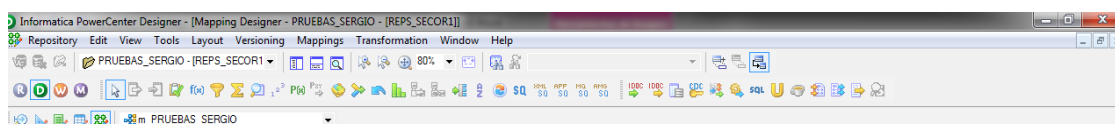


Ilustración 54: Barra de herramientas designer.

Pasaremos a detallar las funcionalidades más frecuentemente usadas en los mapeos de los desarrollos. Esta descripción la dividiremos en **aspectos funcionales de la aplicación** (en ellos veremos que posibilidades a nivel de aplicación nos ofrece la herramienta, por ejemplo, importar objetos, buscar determinados campos o tablas dentro de un mapeo, crear-borrar mapeos, crear transformaciones, etc.) y en **aspectos conceptuales de la aplicación** (veremos cuales son las funcionalidades de cada uno de los componentes mas utilizados en la herramienta y como se usan los mismos).

### 2.6.2.1 Búsqueda en el workspace.

Ésta es una de las funcionalidades que comentábamos sobre un aspecto funcional de la aplicación. Con ella se puede buscar un campo o una tabla completa dentro de todos los componentes del mapeo que tengamos abierto.

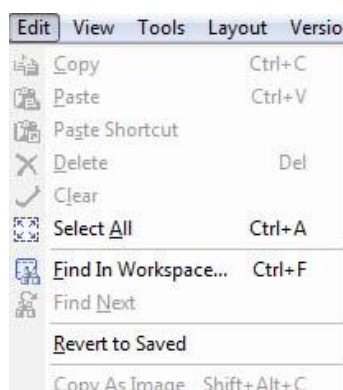


Ilustración 55: Menú edit.

Se accede a ella desde la barra de herramientas en el menú “edit/ Find in workspace” al pulsar sobre esta opción saldrá una ventana con el siguiente aspecto.

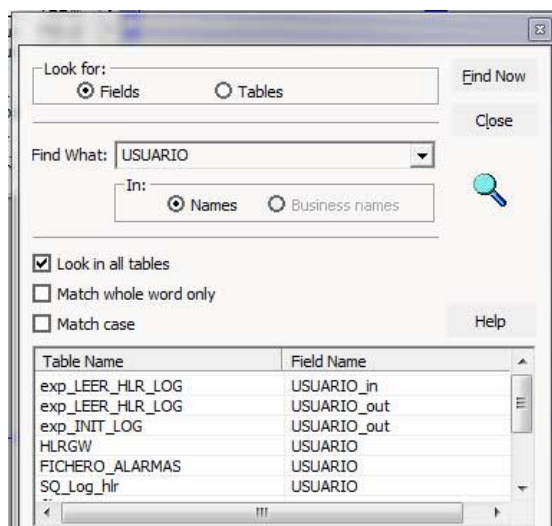


Ilustración 56: búsqueda en el workspace.

En primer lugar se marca el objeto de la búsqueda, si será un campo o una tabla. Después en el cuadro de texto se introduce el nombre del campo o tabla a buscar y se puede indicar que mire en todas las tablas de ese mapeo, que busque solo los componentes en los que coincida el nombre buscado literalmente, esta diferencia no se hace entre mayúsculas y minúsculas. Si se marca la última opción el buscador si hará distinción entre las mayúsculas y minúsculas.

Al pulsar en el botón de buscar aparecerá un listado con todos los componentes en los que aparece el texto indicado o que contengan ese texto dentro del mapeo concreto que se tenga abierto.

### 2.6.2.2 Organización de componentes.

Desde el menú "layout/Arrange.." se puede elegir entre tres opciones de organización de los componentes del mapeo.

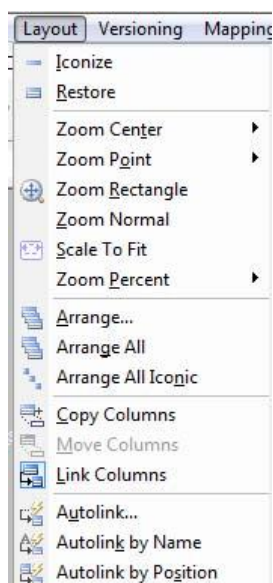
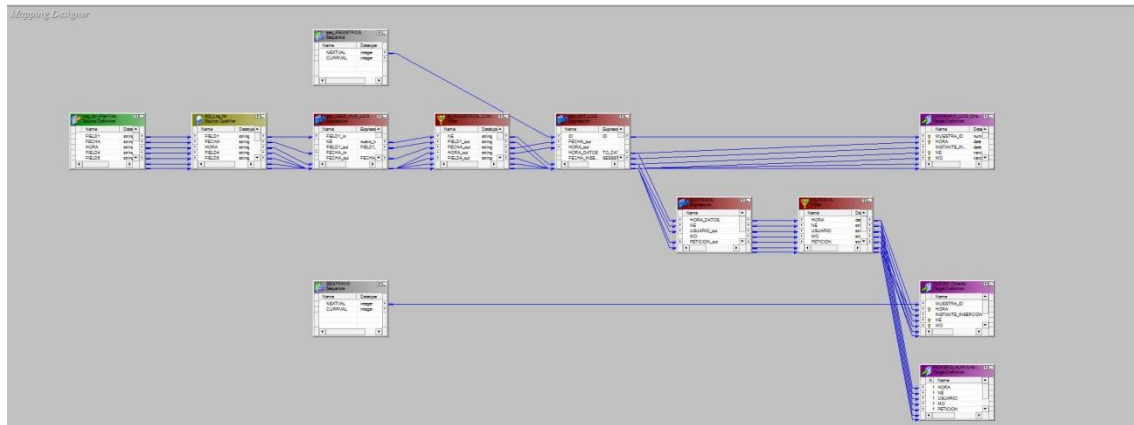


Ilustración 57: menú layout.

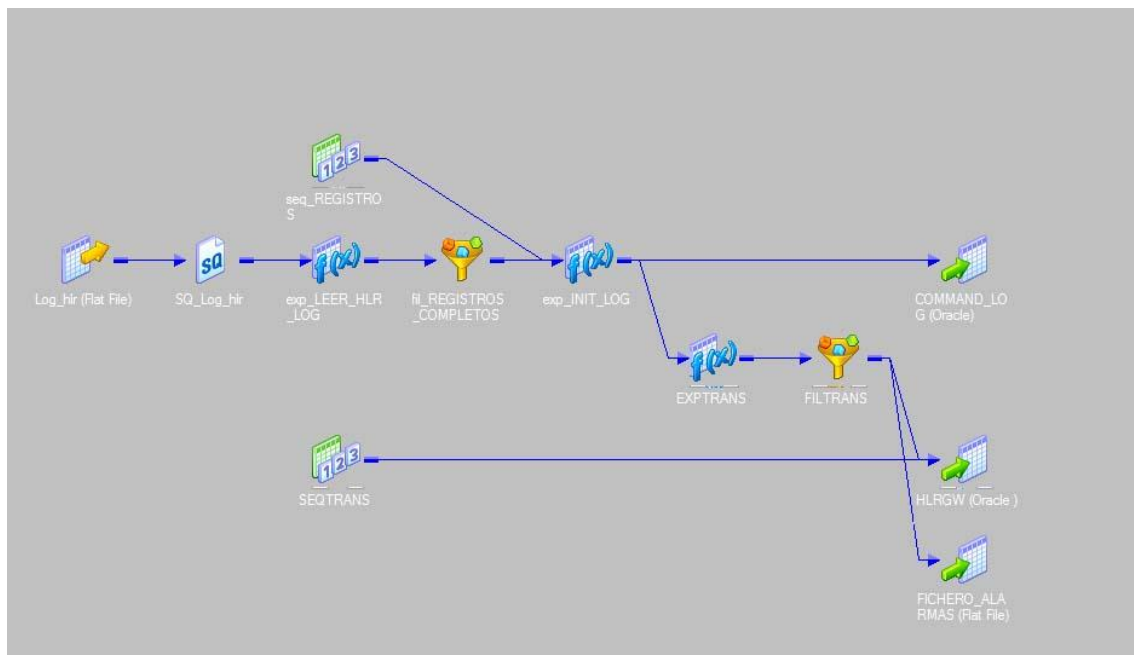
La primera opción nos permite seleccionar dentro de un mapeo los distintos destinos que tengamos y así poder ver y ordenar que componentes sean usados en el mapeo que escriben en ese destino.

Si pulsamos la segunda opción “arrange all”, los componentes se organizan en su aspecto desplegado, es decir, viéndose todos sus campos y sus atributos.



**Ilustración 58: Vista de la ordenación desplegada.**

Por último, si pulsamos la opción “arrange all iconic” los componentes se ordenan presentando su forma compacta en forma de iconos.



**Ilustración 59: Vista de la ordenación con iconos.**

### 2.6.2.3 Auto enlazado de componentes.

En el menú “Layout” ([ilustración 57](#)) se encuentra la opción de “autolink..”, en ella podemos realizar el enlazado de los puertos de distintos componentes los componentes con tal solo indicar que dos componentes queremos enlazar la aplicación ya se encarga de enlazarlos de manera automática atendiendo a los nombres de los puertos de cada componente o simplemente atendiendo a sus posiciones dentro de los dos componentes.

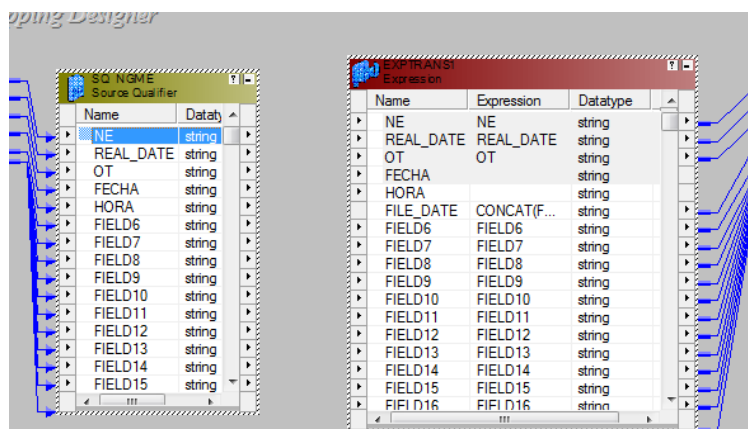


Ilustración 60: Componentes de un mapping.

Aquí se pueden ver dos componentes, que son las distintas cajas que se aprecian,( cada caja es un componente o transformación distinta, hay varios diferentes y cada uno ofrece la posibilidad de implementar varias funcionalidades , con ellos son con lo que trabaja principalmente la herramienta. Dentro de cada componente se encuentran los puertos, que son los campos que corresponderían a la tabla de bbdd o a un fichero indicado.) no estan enlazados sus puertos por lo que los datos no se trasladarian de uno a otro y no se podrian transformar hasta llegar al componente de destino.

Esta funcionalidad permite que el desarrollador enlace los distintos puertos de uno y otro componente sin tener que estar pendiente de si son correctos o ir uno a uno comprobandolo. Para ello clickamos en el menú “Layout / autolink” y nos saldrá la siguiente ventana en la que se podra elegir cual es la transformación de origen y cual es la transformación de destino con la que queremos enlazar sus puertos. Posteriormente se tiene que indicar si queremos que se enlacen los puertos atendiendo a los nombres que hay en cada una de las transformaciones o si simplemente queremos que se atienda a la posicion de estos. Si elegimos la segunda opción y hay mas puertos en una transformacion que en otra, se iran uniendo los distintos campos de una y otra hasta que una de ellas se quede sin mas campos para unir.

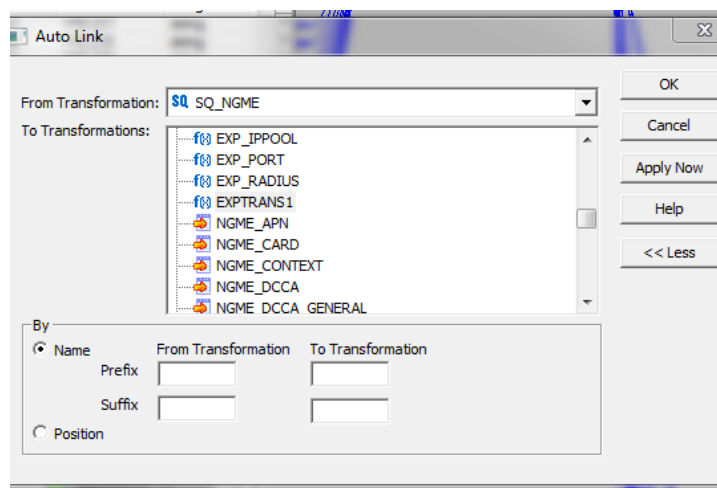


Ilustración 61: ventana auto link.

Al finalizar esta acción los dos componentes quedan con todos sus puertos unidos.

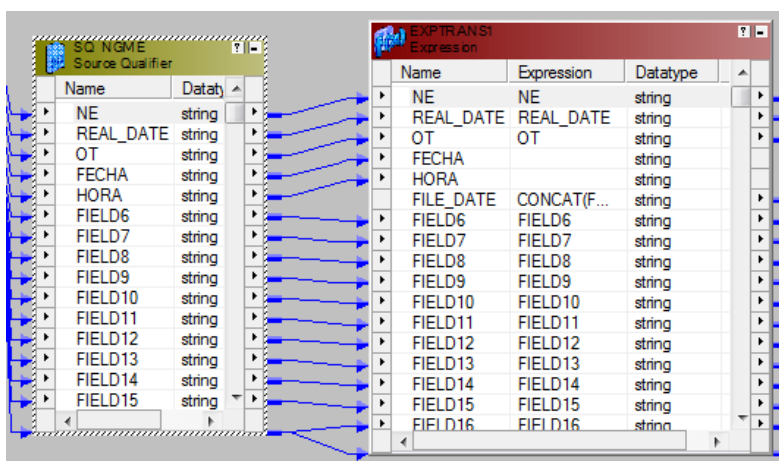


Ilustración 62: Componentes enlazados.

#### 2.6.2.4 Opciones sobre vistas concretas de trabajo.

Este menú variará según la vista que tengamos visible en la aplicación ([vistas sobre la aplicación designer](#)), pudiendo ser éstas como en otros momentos de este documento se ha comentado: editor de orígenes o destinos, editor de transformaciones, editor de mapplets o editor de mappings, como es el caso. En cada una de estas vistas el menú cambiara de nombre adoptando el de la vista en la que nos encontremos situados y nos ofrecerá distintas opciones aplicables a cada uno de los editores.

El menú “Mapping” ofrece varias opciones generales típicas que se pueden aplicar sobre los mapeos, no son tanto funcionalidades en sí mismas, si no mas bien opciones comunes que debe tener cualquier aplicación.

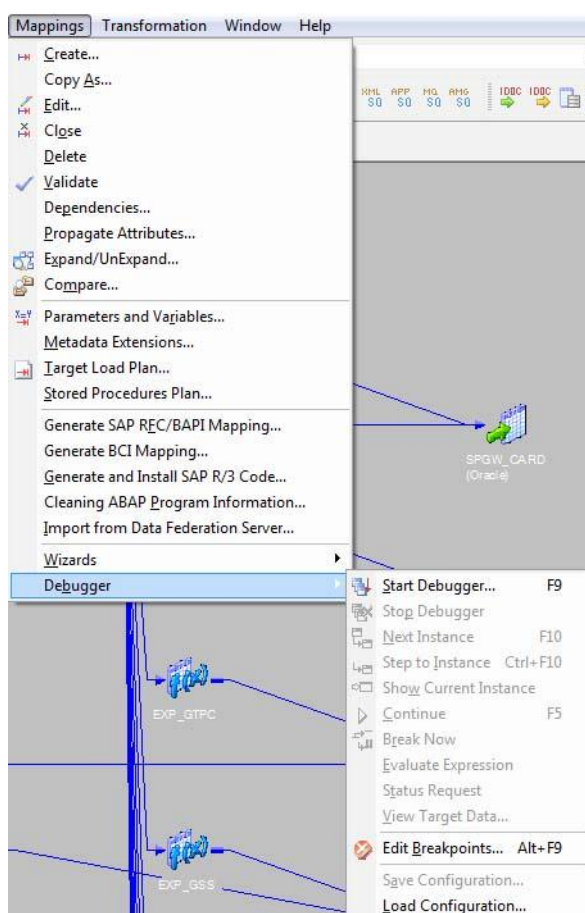


Ilustración 63: Menú Mapping.

En éste menú podemos ver las opciones comunes de crear, copiar, editar, borrar... Además cabe destacar de éste, las opciones de dependencias, propagación de atributos, comparar mapeos, definir parámetros y variables, definir el plan de carga y la opción de debug.



## Vista Mapping.

### Dependencias.

Esta funcionalidad permite, que una vez elegido un mapeo, podamos comprobar cuales son todos sus componentes relacionados y si estos se están reutilizando en algún otro desarrollo del repositorio.

Para ello una vez que se pulsa la opción de dependencias del menú se muestra la siguiente ventana para poder elegir que opciones queremos seleccionar y hasta que nivel de relación queremos que se muestre.

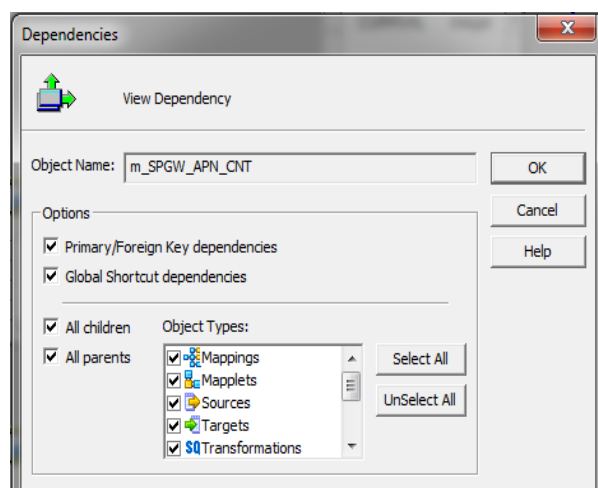


Ilustración 64: ventana de configuración de búsqueda de dependencias.

Se puede seleccionar que nos muestre las dependencias de las claves de las tablas o si hay alguna tabla de acceso directo o “shortcut”.

Una vez seleccionado el nivel de detalle y los componentes que deseamos que nos muestre relación con el mapping que hemos indicado pulsando el botón “OK” se mostrara la siguiente ventana resumen con todas las relaciones.

View Dependencies [Repository: REPS_SECOR1 - Folder: SPGW Object: m_SPGW_APN_CNT]										
Object Name	Group Name	Object Type	V...	Time Stamp	Status	Version Comments	Folder Name	User Name	Host Name	Is Mo...
JNRTRANS		Joiner	1	05/29/2013 14:10:36	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS3		Expression	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS2		Expression	1	05/06/2013 14:13:34	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS		Expression	1	05/06/2013 14:13:34	Active		SPGW	admin_gmv	"UNKNOWN"	No
SEQTRANS		Sequence	1	06/26/2013 13:00:29	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS4		Expression	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
INPUT		Input Tran...	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS2		Expression	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
TABLAS_COLUMN...	SECORDB_Prod	Source Def...	1	05/06/2013 16:58:58	Active		SPGW	admin_gmv	"UNKNOWN"	No
map_COUNTER_S...		Mapplet	1	05/29/2013 14:10:36	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS1		Expression	1	05/06/2013 17:00:02	Active		SPGW	admin_gmv	"UNKNOWN"	No
SQ_SPGW_APN1		Source Qu...	1	07/24/2013 13:13:06	Active		SPGW	admin_gmv	"UNKNOWN"	No
FILTRANS		Filter	1	05/06/2013 14:13:34	Active		SPGW	admin_gmv	"UNKNOWN"	No
seq_SPGW_APN_C...		Sequence	1	06/26/2013 13:00:29	Active		SPGW	admin_gmv	"UNKNOWN"	No
ESTADISTICAS_ME...	SECORDB_Prod	Source Def...	1	05/06/2013 16:58:58	Active		SPGW	admin_gmv	"UNKNOWN"	No
SQ_SPGW_APN		Source Qu...	1	07/24/2013 12:47:48	Active		SPGW	admin_gmv	"UNKNOWN"	No
OUTPUT		Output Tra...	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
SPGW_APN_CNT_D		Target Def...	1	05/28/2013 10:00:12	Active		SPGW	admin_gmv	"UNKNOWN"	No
AGGTRANS		Aggregator	1	05/06/2013 14:13:34	Active		SPGW	admin_gmv	"UNKNOWN"	No
AGGTRANS		Aggregator	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
SPGW_APN_CNT		Target Def...	1	07/24/2013 11:45:11	Active		SPGW	admin_gmv	"UNKNOWN"	No
SPGW_APN	SECORDB_Prod	Source Def...	1	07/24/2013 11:02:22	Active		SPGW	admin_gmv	"UNKNOWN"	No
INPUT1		Input Tran...	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
m_SPGW_APN_CNT		Mapping	1	07/24/2013 13:13:06	Active		SPGW	admin_gmv	"UNKNOWN"	No
EXPTRANS		Expression	1	05/06/2013 17:00:02	Active		SPGW	admin_gmv	"UNKNOWN"	No
JNRTRANS1		Joiner	1	05/08/2013 14:35:38	Active		SPGW	admin_gmv	"UNKNOWN"	No
SQ_ESTADISTICAS...		Source Qu...	1	05/06/2013 17:00:02	Active		SPGW	admin_gmv	"UNKNOWN"	No
wf_SPGW_CNT_re...		Workflow	1	05/28/2013 13:21:54	Active		SPGW	admin_gmv	"UNKNOWN"	No
wf_SPGW_CNT		Workflow	1	05/28/2013 13:20:35	Active		SPGW	admin_gmv	"UNKNOWN"	No
s_m_SPGW_APN...		Session	1	05/28/2013 13:18:25	Active		SPGW	admin_gmv	"UNKNOWN"	No

Ilustración 65: Tabla de elementos relacionados de un mapeo.

### Propagación de atributos.

Esta funcionalidad permite al desarrollador cambiar las propiedades de los distintos campos de cada una de las transformaciones y poder propagar estos cambios a las sucesivas transformaciones a partir de la seleccionada.

Nos permite propagar los atributos de unos pocos campos de una transformación o de todos los campos de la transformación si fuera necesario. Para ello tenemos que pinchar en la opción propagar atributos que se presenta en la barra de herramientas en la opción “Mapping/ Propagate atributes”.

Una vez realizados los cambios sobre la transformación de los campos que sean necesarios cambiar sus definiciones se seleccionan dichos campos y cuando se clickea en la opción anteriormente indicadas se muestra la siguiente ventana.

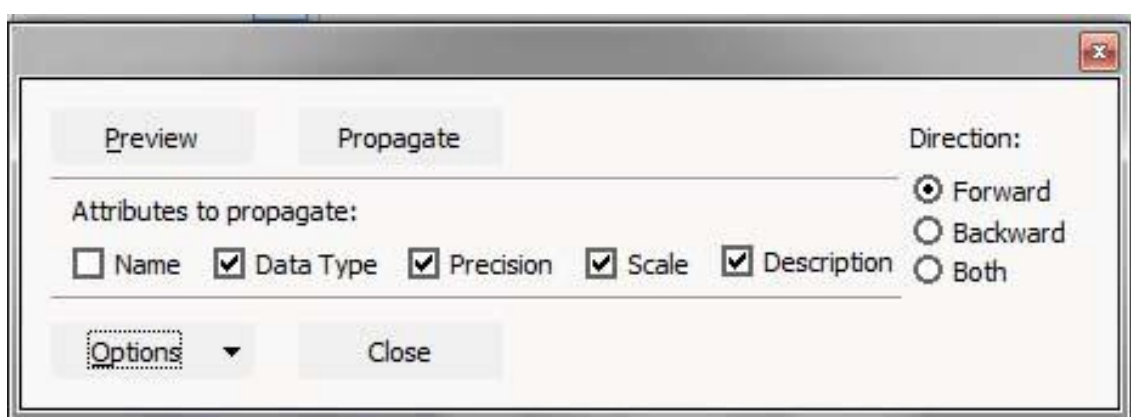


Ilustración 66: Propagación de atributos.

Aquí debe el desarrollador seleccionar cuales son las propiedades de los atributos que se han modificado y que se desean propagar al resto de elementos de ese flujo de datos. Puede elegir entre las distintas opciones como son: el nombre del campo, el tipo de datos, la precisión (longitud del campo), la escala ( en caso de ser un numérico indicaría cuales son los decimales que puede llevar) y por último la descripción del campo. También puede indicar si desea que los cambios se apliquen en un sentido u otro del flujo de datos o en ambos.

Una vez seleccionados los parámetros que se desean propagar se pulsa sobre el botón “propagate” y la aplicación automáticamente realiza los cambios sobre todo el flujo de datos correspondiente a esa rama del mapeo.

### Comparador de mapeos.

Esta opción permite al desarrollador comparar dos mapeos y poder encontrar sus diferencias o similitudes. Para ello se abre un mapeo que se desee comparar y pulsamos sobre la opción “Mapping/Compare” de tal manera que aparecerá la siguiente ventana.

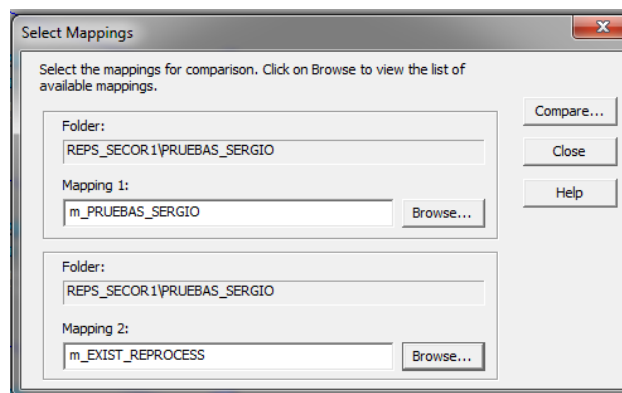


Ilustración 67: Comparador de mapeos.

Aquí aparecerá seleccionado por defecto el mapeo que se tiene abierto, aunque este podrá ser modificado y además tendremos que seleccionar un mapeo del mismo directorio.

**Nota aclarativa:** En esta opción se comparan mapeos del mismo directorio para comparar mapeos o cualquier otra estructura diferente recordamos que existía una función comparar en la aplicación repository manager y además existe análogamente una función comparar en la aplicación workflow manager que comparará workflows.

Una vez seleccionados ambos mapeos que se deseen comparar se pulsa sobre el botón “compare” y se mostrará una nueva ventana con el siguiente resultado.

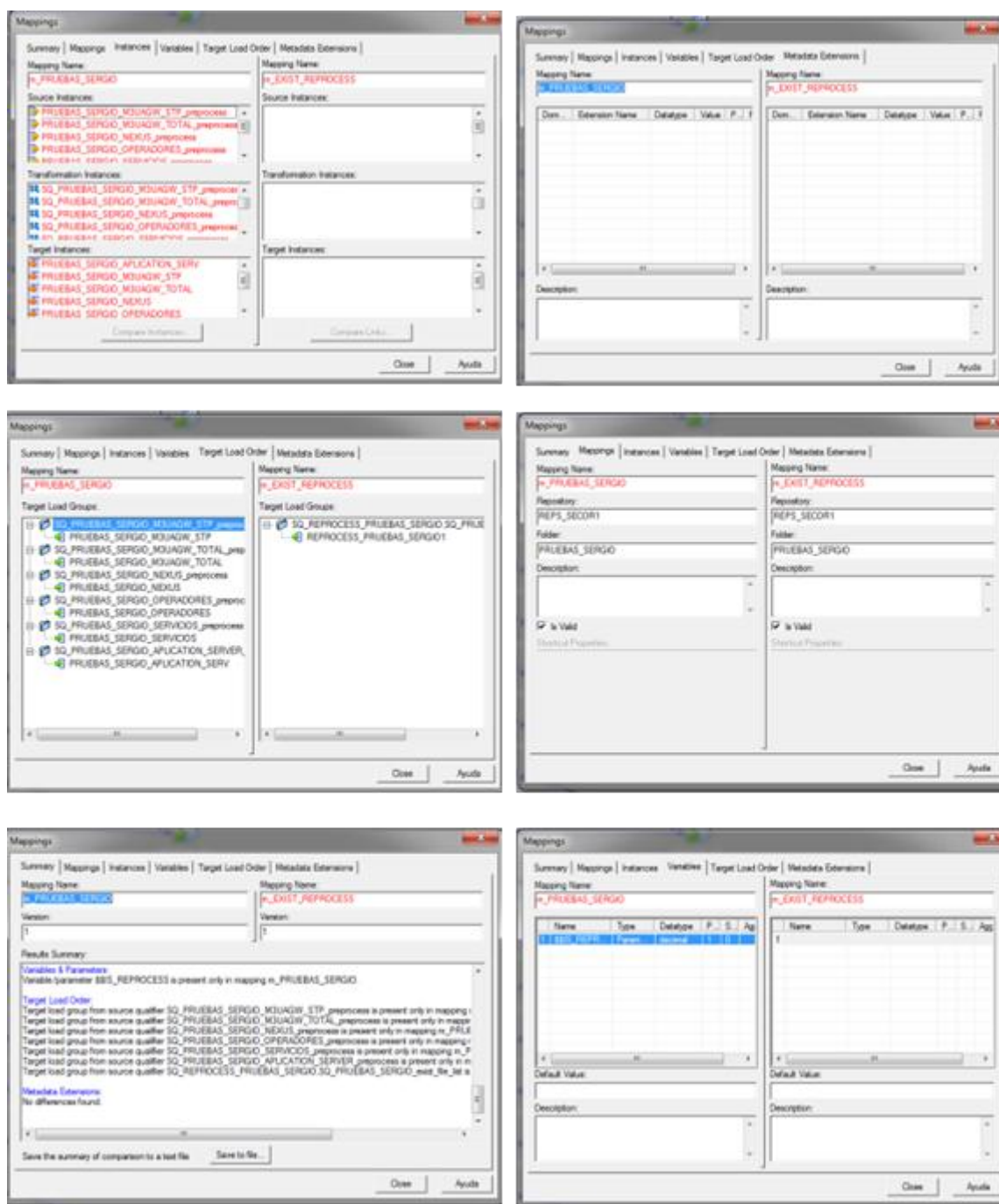


Ilustración 68: Resultado del comparador de mapeos.

En esta ventana se presenta la información detallada de los resultados obtenidos de las comparaciones realizadas. En primer lugar muestra una pestaña con el resumen de los resultados obtenidos mostrando cada mapeo comparado, la versión de los mismos y un cuadro de texto con todos los resultados. Además tiene unas pestañas que muestran un detalle mayor acerca de distintos aspectos comparados de los mapeos: Mappings, instancias, variables, tablas de destino de datos y metadatos.

### Parámetros y variables.

Una de las opciones más usadas y necesarias en esta aplicación es el uso de variables auxiliares o de parámetros pasados a través de ficheros. Estos valores se le pasaran a la aplicación en el momento de su invocación pero para poder ser usados deben de estar definidos en el mapeo donde se vayan a usar. Los parámetros y las variables se definen para cada mapeo, no es posible definir un parámetro o variable genérica ya que en el fichero de parámetros hay que indicarle que sesión usara cada parámetro. Se podría usar el mismo nombre de parámetro y que este tuviera el mismo valor pero es imprescindible que en cada mapeo se indique que parámetros o variables recibe ese mapeo en concreto.

Esta opción se configura pulsando en el menú “Mapping/Parameters and variables” y muestra al desarrollador la siguiente ventana.

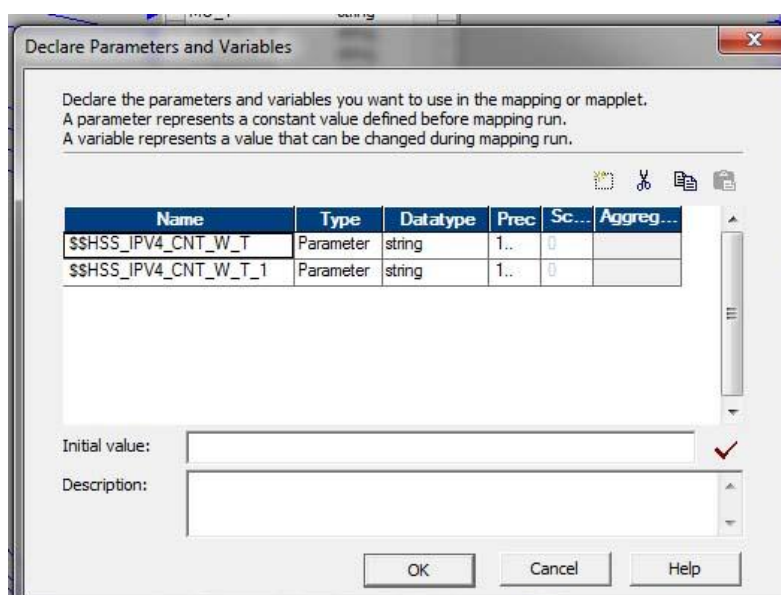


Ilustración 69: Parámetros y variables.

Pulsando sobre el icono que aparece en la parte superior derecha con forma de un cuadrado con una estrella en su parte superior se creará un nuevo registro en blanco y en el podremos introducir el nombre del parámetro o variable, siempre debe llevar el nombre de estos los símbolos “\$\$” para que la aplicación los reconozca como parámetros, después tendremos que definir si será un parámetro o una variable, el tipo de datos que será, la longitud del parámetro y se le puede indicar un valor inicial por defecto.

Posteriormente una vez salvados estos parámetros o variables podrán ser usados en cualquier transformación en el apartado expresión, ahí aparecerá una ventana con un apartado que contendrá todos los parámetros y variables definidos en el mapeo.

Plan de carga.

Esta opción permite al desarrollador dado un mapeo con distintas ramas o flujos de ejecución dentro de un mismo mapeo (mapping) elegir en qué orden quiere que se vayan ejecutando los distintos flujos de carga.

Para ello se sitúa en la opción de la barra de herramientas “Mapping/target load plan” y se mostrará la siguiente ventana.

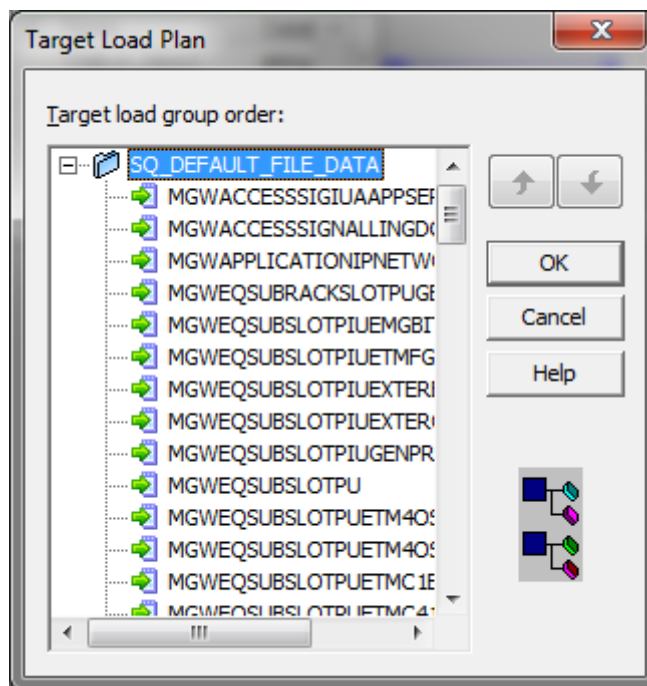


Ilustración 70: Plan de carga.

En dicha ventana se mostrarán todos los destinos que contenga el mapeo seleccionado, el desarrollador podrá seleccionar cualquier destino que elija y podrá ir situando su orden de carga desplazando estos destinos hacia arriba o hacia abajo en la lista que aparece en la imagen con las flechas que aparecen en la parte derecha de la ventana. El flujo de carga se irá ejecutando según el orden establecido en la lista comenzando por el primero hasta llegar al último.

Transformaciones.

Cada uno de los componentes o “cajas” que hay en un mapeo se pueden crear utilizando dos maneras para acceder hasta ellos: la primera consistiría en pulsar sobre el icono directo que tienen las transformaciones más usadas en la barra de iconos y la segunda consistiría en ir a la opción “Transformation” en la barra de herramientas.

Una vez pulsada esta opción nos saldrá una ventana con el siguiente aspecto y ésta contendrá todas las transformaciones predefinidas y disponibles en la aplicación.

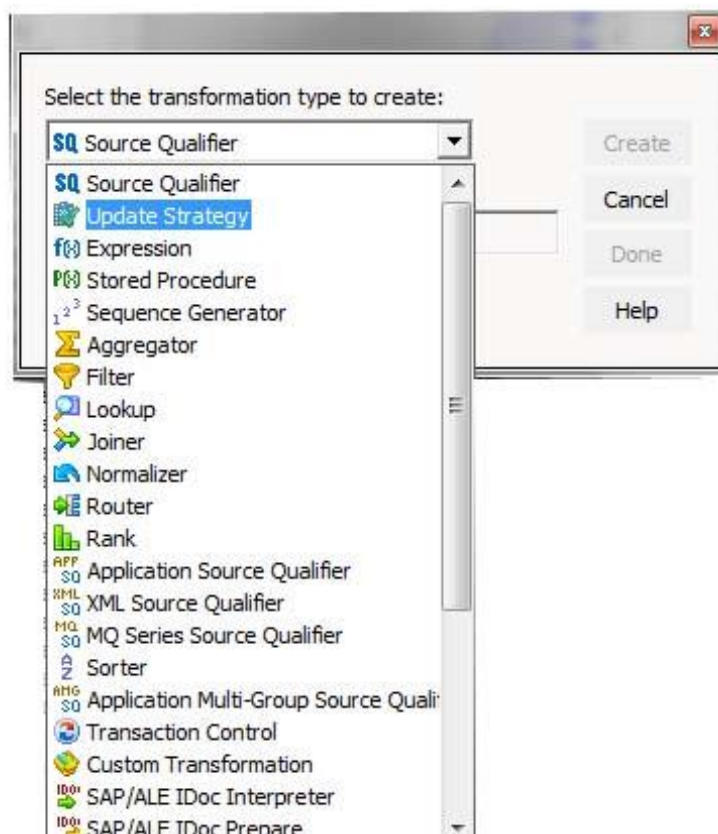


Ilustración 71: ventana de transformaciones.

Una vez mostrada esta ventana se deberá seleccionar la transformación que sea necesaria y esta se mostrará en el mapeo, ahora el desarrollador tendrá que configurar dicha transformación como haría con otra transformación cualquiera, tendrá que añadirle los campos que desee cargar y realizar las operaciones que le sean necesarias dentro de la transformación.



## Vista Orígenes.

Importar desde varios orígenes.

Para desarrollar un mapeo de carga es necesario en primera estancia definir cual serán los orígenes de los cuales recogerá los datos nuestro proceso. Para ello podemos realizar la definición de dichos orígenes importando su estructura desde distintas fuentes. Estas fuentes serán las siguientes:

- Importar desde una bbdd: Aquí podremos definir nuestro origen cuando éste sea una tabla de una bbdd a la cual tengamos conexión. Para ello iremos al menú “sources/import from database...”.

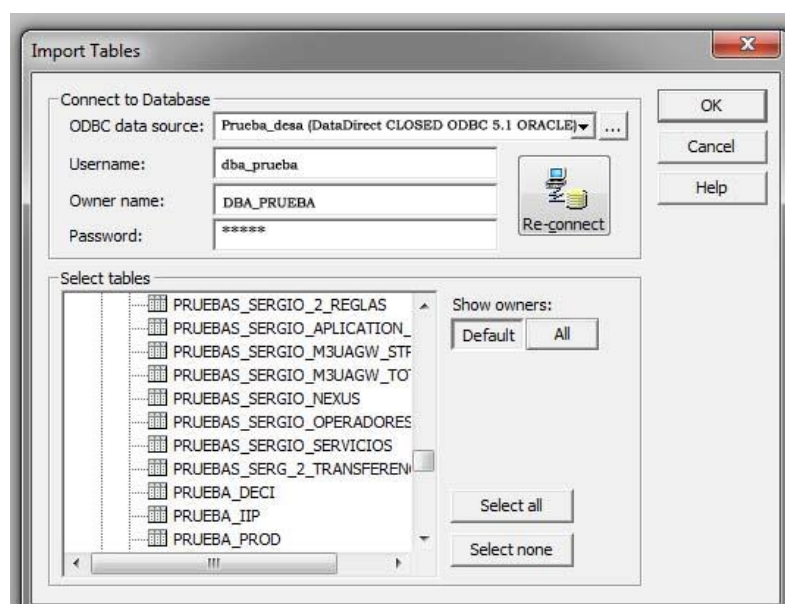


Ilustración 72: Importar tabla de origen.

Una vez conectados a la bbdd, aparecerá un listado con todas las tablas y vistas sobre las cuales tenga permisos ese usuario. Se seleccionaran una o varias tablas que deseemos importar como orígenes y pulsaremos el botón “OK”. Acto seguido se cerrará esta ventana y aparecerán en la vista de orígenes de la aplicación designer, todas las tablas que hayamos seleccionado.

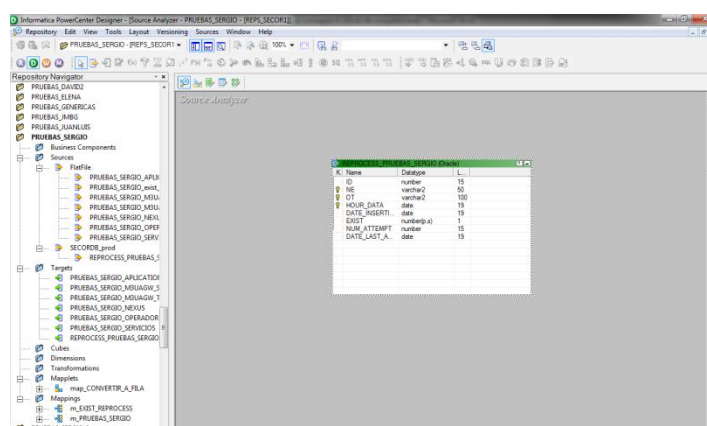
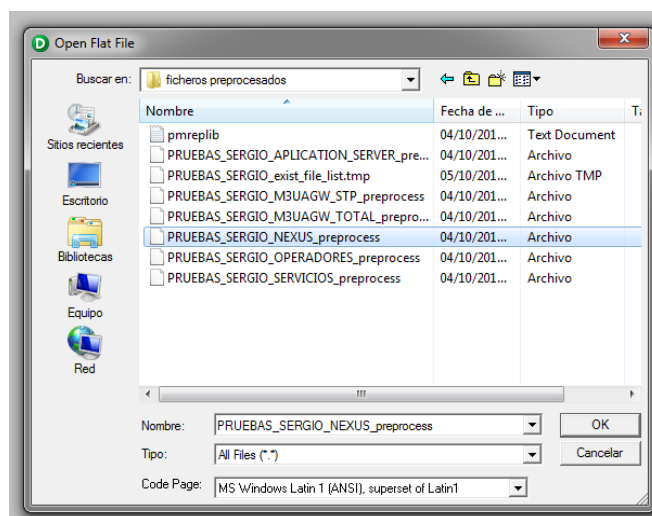


Ilustración 73: Vista de orígenes importados.

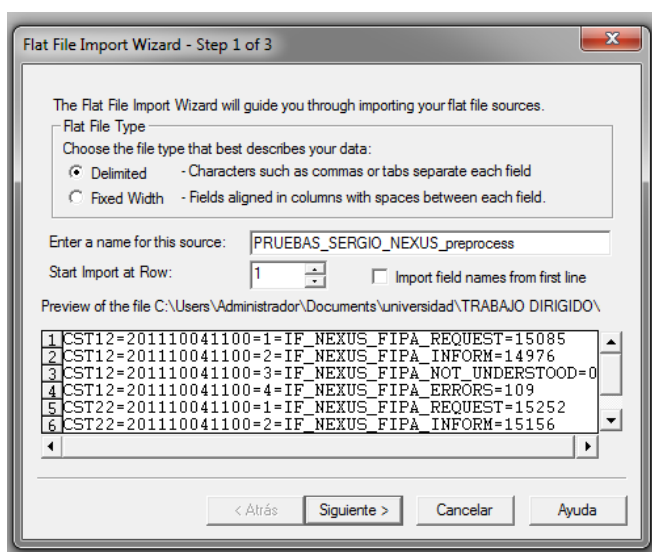


- Importar desde un fichero plano: Con esta opción se podrá definir un origen a partir de un fichero plano. Se realizará pulsando la opción “Import from flat file” del menú Sources.



**Ilustración 74: Importar un origen de fichero, paso 1.**

En primer lugar al pulsar la opción de importar un origen desde un fichero plano, saldrá una ventana en la cual habrá que seleccionar el fichero que queremos que actúe como origen de nuestro mapeo.



**Ilustración 75: Importar un origen de fichero, paso 2.**

Al seleccionar el fichero se abrirá un asistente en el cual tendremos que elegir si el fichero es un fichero con caracteres delimitadores de campos o si en su defecto son campos con una longitud fija de caracteres, en esta primera ventana también se permite elegir el nombre que queremos dar al origen y también si hubiera alguna línea de cabecera indicar cuantas líneas vamos a descartar de la cabecera del fichero cuando procesemos un fichero de ese tipo.

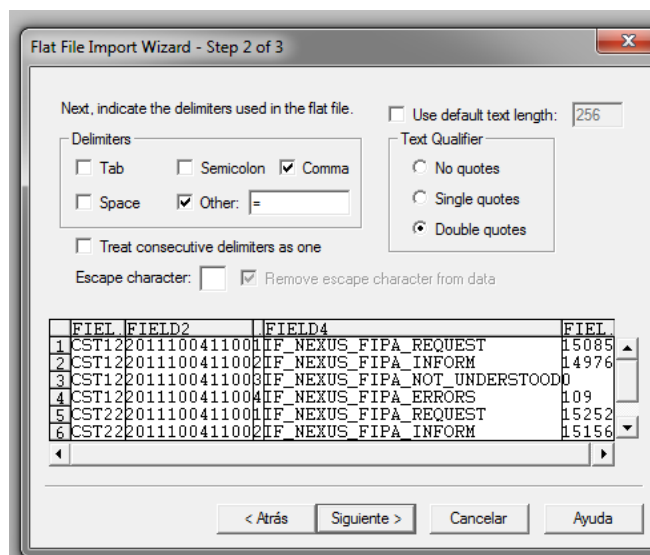


Ilustración 76: Importar un origen de fichero, paso 3.

En la siguiente ventana tendremos que indicar, en caso de que el fichero que se haya escogido sea delimitado por algún carácter, cual será el carácter que realiza las funciones de separador de campos, podemos elegir uno o varios y en caso de no ofrecerse el carácter separador en los delimitadores predefinidos podremos seleccionar la opción “Other” e indicarle el/los caracteres que nos sean necesarios.

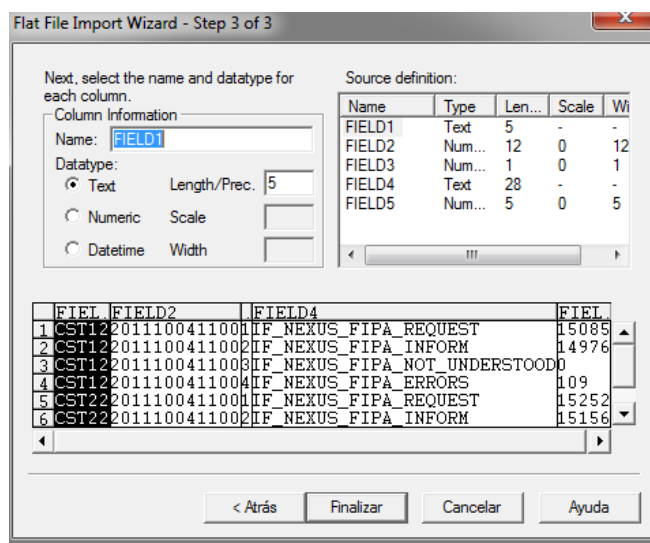
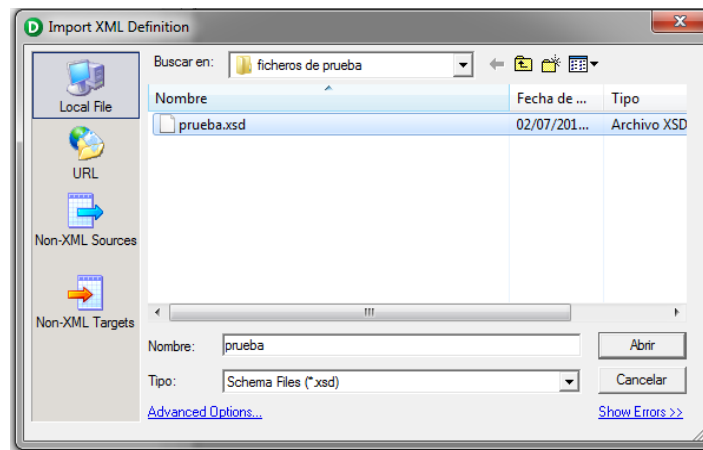


Ilustración 77: Importar un origen de fichero, paso 4.

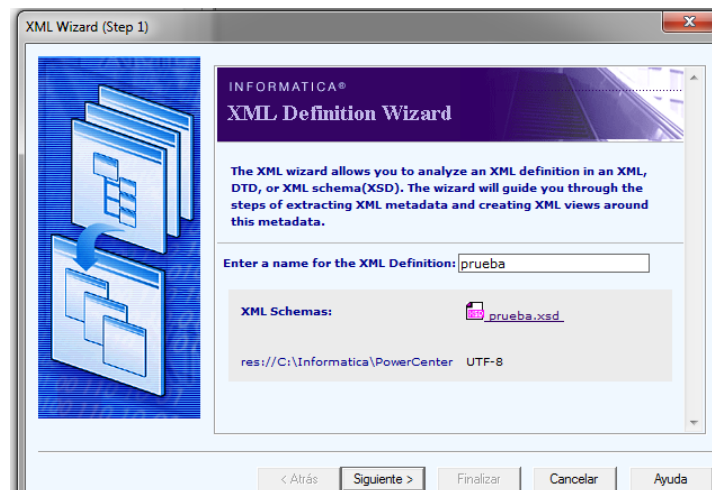
Una vez seleccionado el delimitador tendremos que definir los nombres y los tipos de los campos establecidos, por defecto te recoge la aplicación el tipo y la longitud de los campos pero el desarrollador puede modificar según sus necesidades estos valores. Al finalizar este asistente tendremos creado el nuevo origen.

- Importar desde un XML: Pulsando en el menú Sources “import XML definition” se podrá definir un origen desde un fichero XML o desde un XSD.



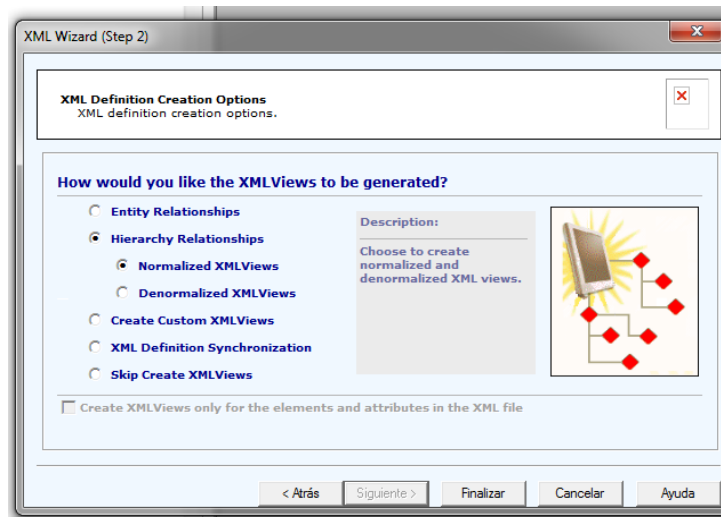
**Ilustración 78: Importar origen desde xml, paso 1.**

Al seleccionar la opción de importar un origen desde un fichero xml, aparece una ventana para seleccionar el fichero del cual queremos extraer la estructura del origen que usaremos.



**Ilustración 79: Importar origen desde xml, paso 2.**

Una vez seleccionado el fichero, aparecerá un asistente, en esta primera ventana aparecerá el nombre del fichero que hemos seleccionado y que tipo de fichero es, si es xml o xsd.



**Ilustración 80: Importar origen desde xml, paso 3.**

En la siguiente ventana se puede elegir que tipo de fichero xml queremos que nos genere si es un fichero de entidad relación, si queremos un fichero normalizado, desnormalizado... Una vez elijamos que tipo de fichero será el que usaremos como origen pulsamos en finalizar.

Name	XPath	Datatype	L...
mdc (X_mdc)			
XPk_mdc		xsd:integer	
ffv	/mfh/ffv	xsd:string	infi...
sn	/mfh/sn	xsd:string	infi...
st	/mfh/st	xsd:string	infi...
vn	/mfh/vn	xsd:string	infi...
cbt	/mfh/cbt	xsd:string	infi...
ts	/mfh/ts	xsd:string	infi...
md (X_md)			
XPk_md		xsd:integer	
FK_mdc		xsd:integer	
neun	/neid/neun	xsd:string	infi...
nedn	/neid/nedn	xsd:string	infi...
nesw	/neid/nesw	xsd:string	infi...
mts	/mi/mts	xsd:string	infi...
go	/mi/go	xsd:short	
mt (X_mt)			
XPk_mt		xsd:integer	
FK_md		xsd:integer	
mt		xsd:string	infi...
mv (X_mv)			
XPk_mv		xsd:integer	
FK_md0		xsd:integer	
moid	/moid	xsd:string	infi...
sf	/sf	xsd:string	infi...
r (X_r)			
XPk_r		xsd:integer	
FK_mv		xsd:integer	
r		xsd:string	infi...

**Ilustración 81: Definición del origen XML.**

Ahora aparecerá en nuestra área de trabajo en la vista de orígenes, un nuevo origen basado en un xml con el siguiente aspecto. Aun así este fichero se podrá editar desde esta vista si fuera necesario.

- Hay otras fuentes de las cuales importar los orígenes, aunque no son tan utilizadas en el día a día paso a mencionarlas en este punto, estas otras opciones serían: importar Cobol, SAP, PeopleSoft, Siebel, PowerExchange...

## Creación de nuevos orígenes.

Se pueden crear nuevos orígenes a partir de nada, en este caso el desarrollador definirá que tipo de origen va a utilizar y definirá cada uno de sus campos indicando: nombre, tipo de datos y tamaño.

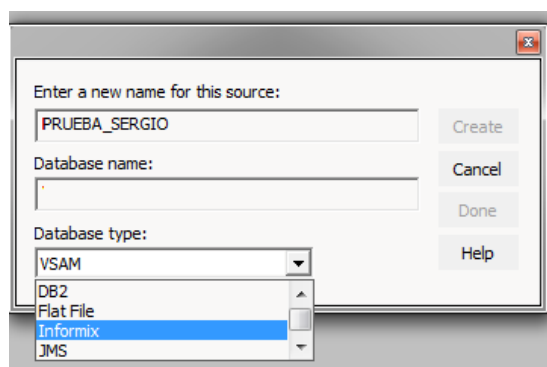


Ilustración 82: Creación de un origen.

Al pulsar sobre el menú “sources/Create” aparece la ventana que se muestra en la ilustración 82 en la cual se tendrá que indicar el nombre de la fuente, el nombre de la bbdd si este fuera el caso de estar creando una tabla y después indicar en el desplegable que tipo de origen vamos a utilizar. Tras rellenar todos estos datos pulsaremos en el botón “Create”, apareciendo en nuestro espacio de trabajo la nueva fuente creada con todos sus campos vacíos.

Ahora para definir los campos que tendrá esta nueva fuente de datos, haremos doble clic sobre el icono y nos aparecerá una ventana en la cual podremos configurar todos los campos que necesite nuestro origen de datos.

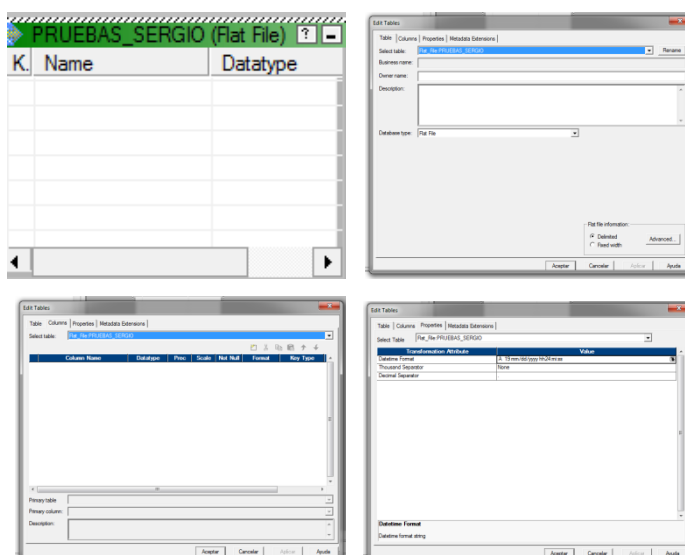


Ilustración 83: Configuración de un origen.

En la primera imagen podemos ver la fuente creada con el asistente vacía. En la segunda imagen se puede ver como una vez que se ha hecho doble clic en el componente nos presenta una primera pestaña con el nombre de “table” en la cual se podrá cambiar el nombre que se le ha indicado al componente así como el tipo de origen que será, al igual que se podía desde el asistente.

En la tercera imagen podemos ver la pestaña columna en ella podremos pulsando el icono con forma de cuadrado discontinuo que hay en la parte superior derecha de la ventana nos permitirá crear todos los campos nuevos que necesitemos. Además en esa barra de herramientas podremos eliminar campos creados, copiar filas, pegarlas y desplazar su orden dentro de nuestro origen de datos.

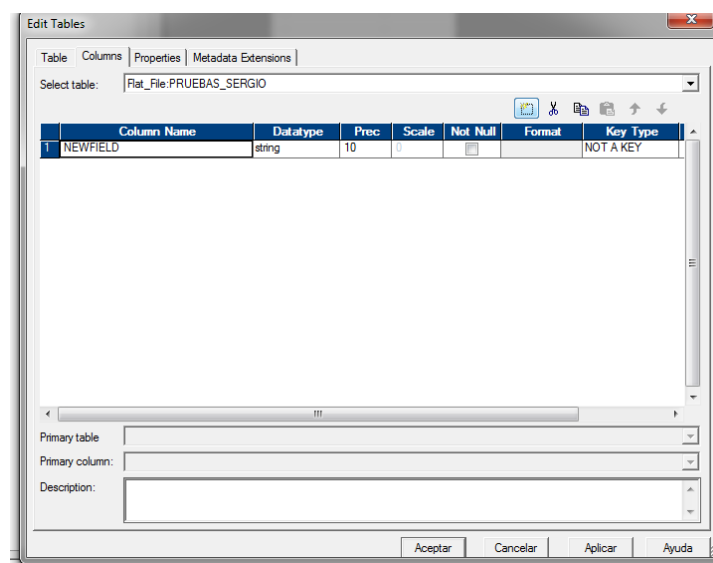


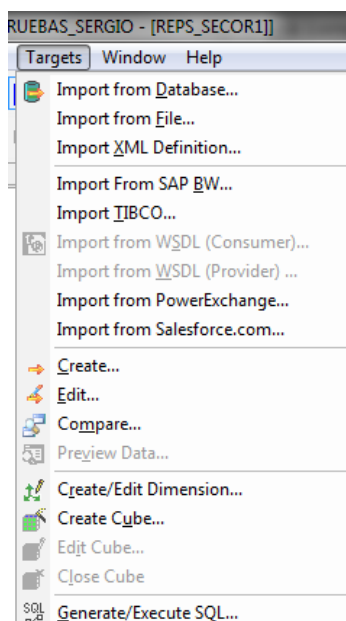
Ilustración 84: Crear campos en un origen.

Por último en la cuarta imagen podemos indicarle el formato de los campos tipo fecha y los separadores de miles y de decimales de los campos numéricos.



### ***Vista Destinos.***

Esta vista ofrece las mismas posibilidades que ofrece la vista de orígenes, permite al desarrollador importar destinos desde una bbdd o desde un fichero, así como crear destinos de manera manual. En la imagen 84 se muestran las opciones posibles para este menú.



**Ilustración 85: Menú destinos.**



### *Vista de Transformaciones.*

En esta vista el desarrollador podrá crear componentes reutilizables para cualquier mapeo a partir de componentes ya prefijados por la herramienta o bien importando distintas funcionalidades programadas en otros lenguajes y que permitiría integrarlas en la aplicación. Por ejemplo desde esta vista se podrían importar procedimientos o funciones PL-SQL definidas en la BBDD o en un código que tengamos en un fichero o bbdd externa. Se podrían importar o crear métodos java, parseadores de xml...

Aquí podremos crear transformaciones encapsuladas, las cuales podremos usar en cualquier mapeo.

Las opciones que permite este menú son las siguientes:

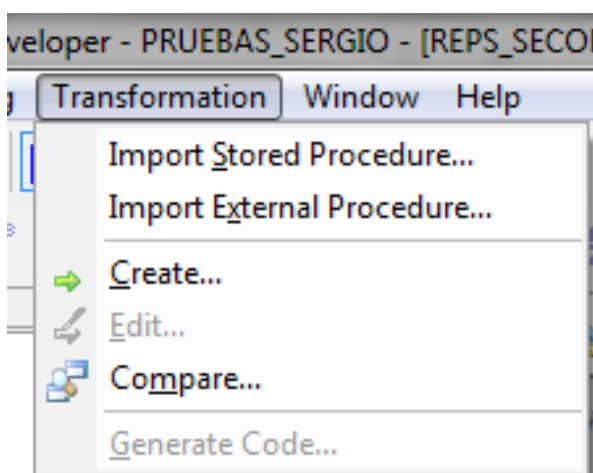


Ilustración 86: Menú transformación.



Importar procedimientos o funciones PL-SQL.

Desde esta vista el desarrollador puede importar procedimientos almacenados en las bdd a las cuales tenga acceso desde la herramienta. Para ello debe pulsar en el menú “Transformation / Import Stored Procedure...”, una vez pulsada esta opción saldrá la ventana de conexión a la bdd y una vez conectados nos aparecerán todos los procedimientos y funciones que haya en la bdd y sobre las que tenga visibilidad nuestro usuario.

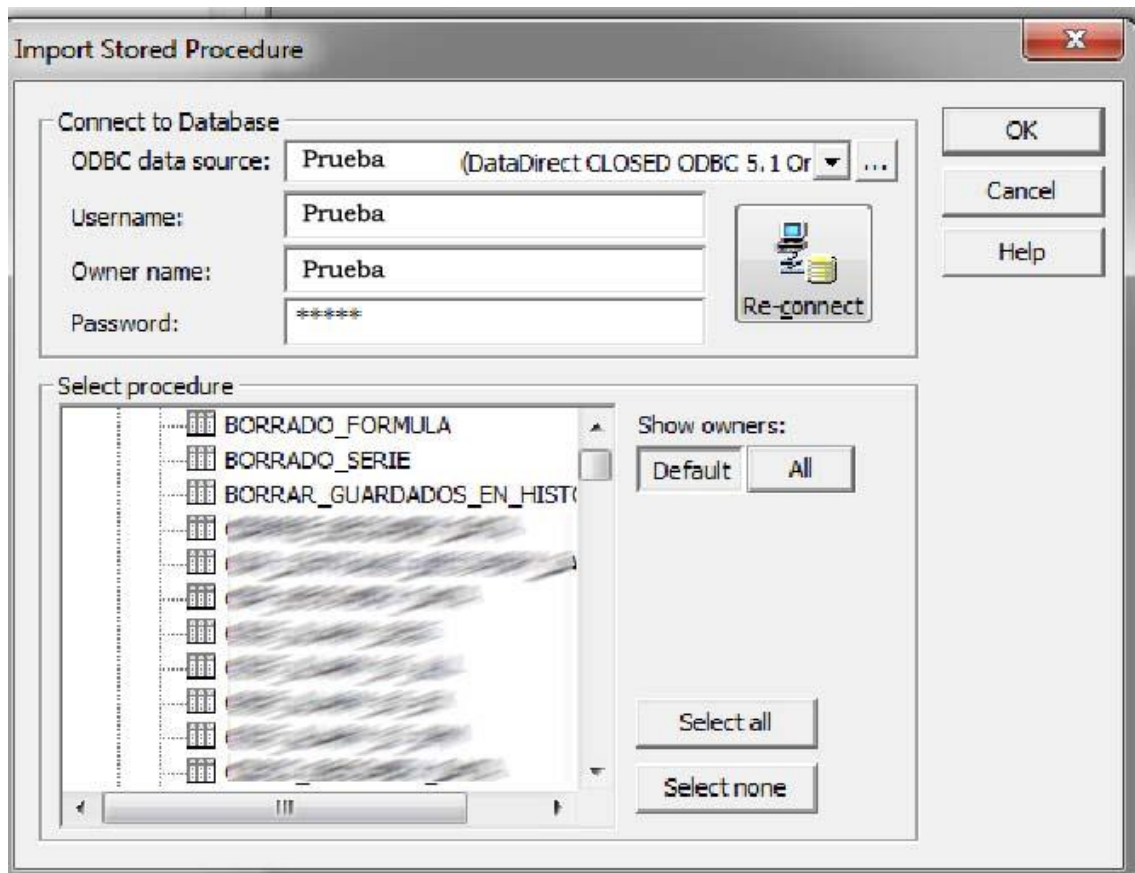
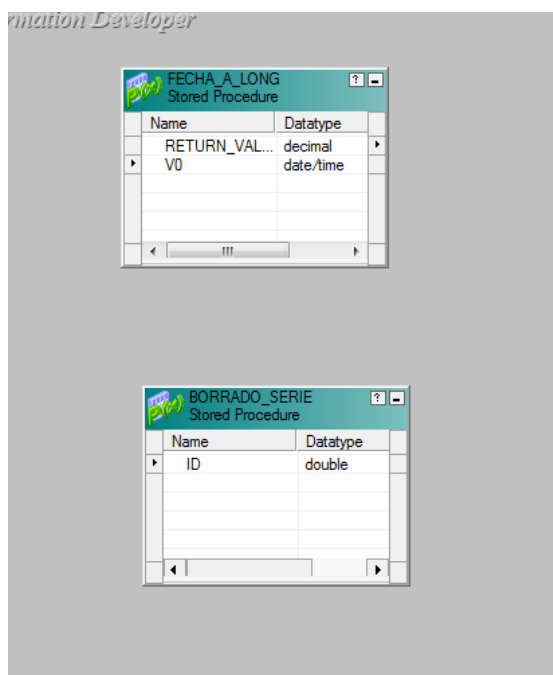


Ilustración 87: Importar procedimientos.

Una vez que hayamos seleccionado el/los procedimientos o funciones que se quieran importar pulsaremos el botón “OK” y aparecerán en nuestro espacio de trabajo de la siguiente manera.



**Ilustración 88: Procedimientos importados.**

Como se puede apreciar en la imagen se crea un componente con el nombre de la función o procedimiento en el cual se tienen unos campos de entrada y unos de salida en caso de que ésta función tuviera que devolver algo. De esta forma podemos ver en la ilustración 88 como la función FECHA\_A\_LONG recibe como entrada una fecha V0 y devuelve un campo return\_value tipo decimal (sería un long, lo que ocurre es que la aplicación no tiene el tipo de datos long, su equivalente en este caso sería el decimal). También podemos ver como en el procedimiento BORRADO\_SERIE solo se le indica como valor de entrada un ID y el procedimiento procederá a borrar esa serie sin devolver nada, el resultado de esta operación se podría ver directamente en bbdd.

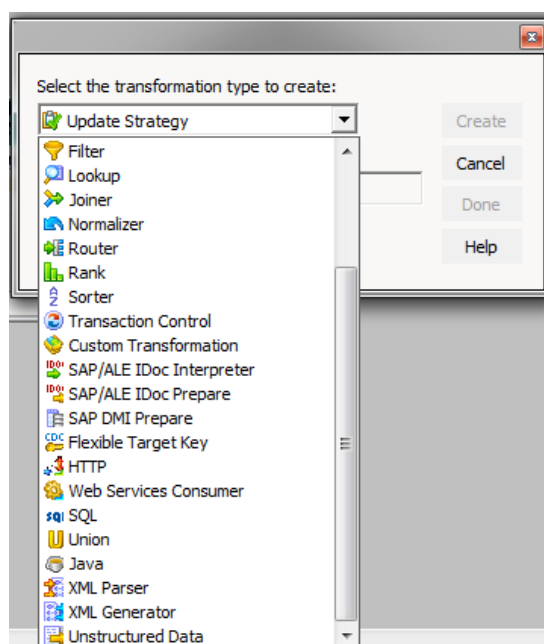
También se pueden importar procedimientos o funciones desde ficheros, para ello se seguirá el mismo procedimiento que en este punto del documento, con la única diferencia que al inicio aparecerá una ventana con un explorador de documentos en la cual elegiremos un fichero con extensión .tlb, .olb, .dll o .ocx, y este cargará los procesos que contenga almacenados, pudiendo elegir entre todos ellos en caso de que tuviera varios.

#### Crear transformaciones.

Con esta opción el desarrollador puede crearse componentes propios que podrá reutilizar en todos los mapeos que realice dentro del mismo directorio, en caso de querer usarlo en otro directorio tendrá que copiarlo o importarlo.

Los elementos a disposición del usuario son todos los predefinidos por la aplicación y estos se podrán personalizar para realizar las funciones que desee el desarrollador. Hay componentes que no se utilizan normalmente pero que podrían resultar interesantes en caso que hubiera

que realizar una transformación en repetidos mapeos y siempre fuera la misma o si hubiera que introducir algún código java o en cualquier otro lenguaje.



**Ilustración 89: Creación de transformaciones.**

Una vez pulsada la opción del menú “Transformation/ Create” aparecerá esta ventana en la cual tendremos que elegir que tipo de transformación queremos realizar e indicarla un nombre, posteriormente pulsaremos en el botón créate y se creará dicha transformación.

Name	Datatype	L...
<b>INPUT</b>		
PRUEBA1_IN	string	10
PRUEBA2_IN	string	10
<b>OUTPUT</b>		
PRUEBA1_OUT	string	10
PRUEBA2_OUT	string	10

**Ilustración 90: Nueva transformación.**

La transformación aparecerá vacía y al igual que en los casos descritos anteriormente tendremos que darle doble clic sobre ella para configurarla indicando los campos de entrada, salida y en este caso las funciones java que queremos que realice según distintos eventos que se produzcan.

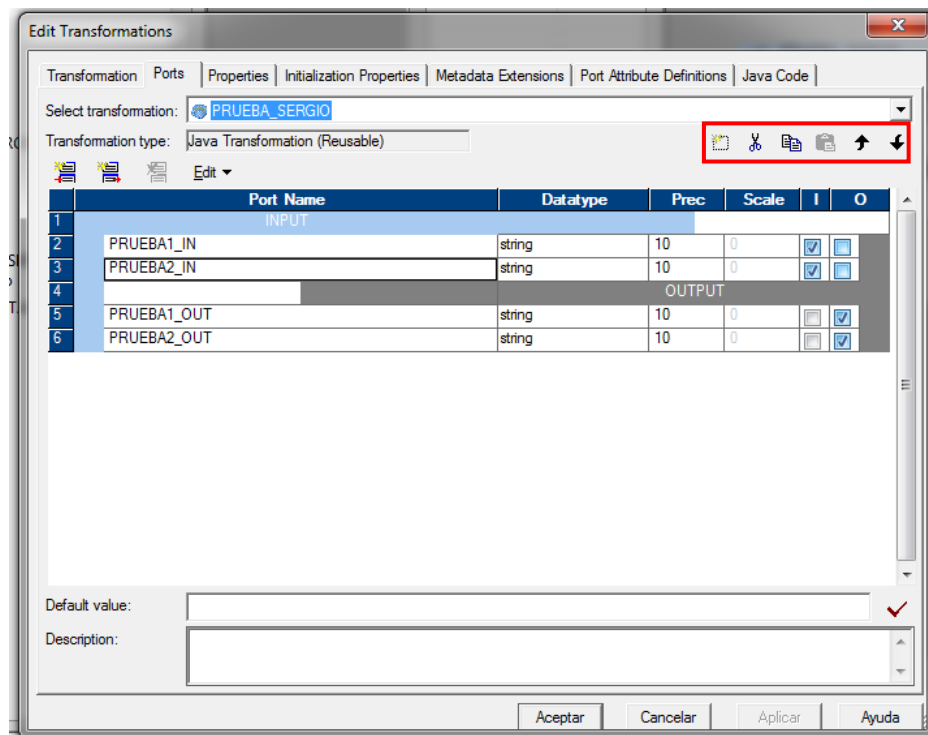


Ilustración 91: Definición de campos entrada y salida.

En esta pestaña podremos configurar cuales serán nuestros campos de entrada y cuales de salida, para ello podremos crear, borrar o cortar, pegar, copiar y desplazar campos dentro de la lista desde los iconos que están resaltados en la parte superior derecha.

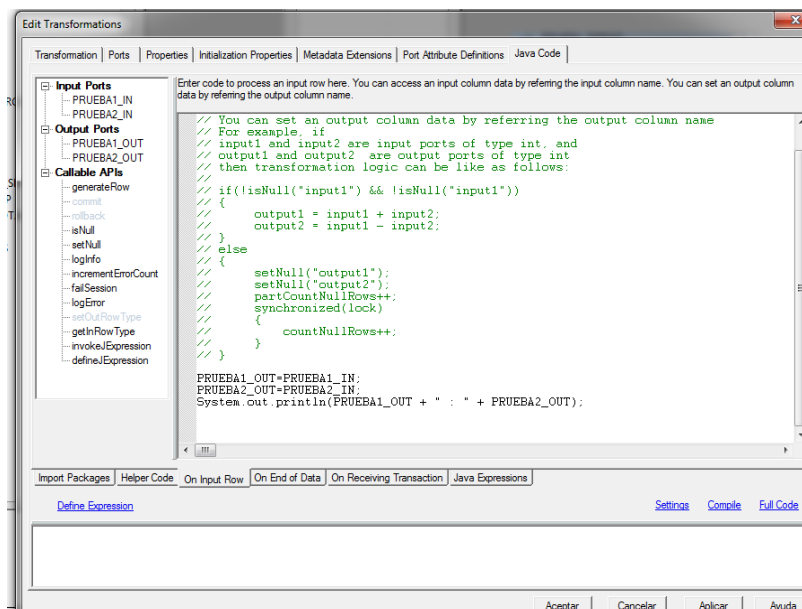


Ilustración 92: Definición del código java a ejecutar.

Una vez indicados los grupos de datos de entrada y salida podremos ir a la pestaña “java code” y podremos incluir las funcionalidades java que precisemos, como se puede ver en la parte inferior de la ventana hay distintas pestañas, estas son para la configuración de la transformación y para incluir código diferente según distintos eventos al recibir un registro, al salir el registro de la transformación o cuando se recibe una transformación. Podremos añadir

el código java que realice la lógica que necesitemos y tendremos a nuestra disposición en la parte izquierda los campos de entrada o salida de la transformación, así como expresiones o cualquier función java que necesitemos.

### *Vista de Mapplets.*

Esta será la vista específica para los componentes llamados mapplets, estos componentes son el equivalente a un método en java, encapsulan un fragmento de la lógica del programa que será reutilizado en varios mappings, en este caso encapsula una parte de la lógica del mapeo y su tratamiento de los datos. En esta vista tendremos las opciones propias para este tipo de componentes.

#### *Crear un mapplet.*

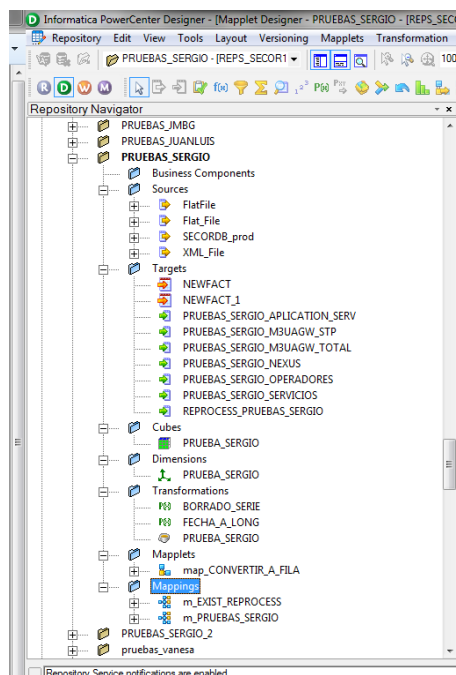
Como se ha comentado en otros puntos del presente documento los mapplets son un fragmento de un mapping que va a ser encapsulado para su reutilización en distintos mapeos.

Para crear un nuevo mapplet hay que ir a la vista de mapplets y pulsaren el menú “mapplets/Create”, Esto nos abrirá una nueva ventana en la cual se nos pedirá introducir el nombre que queramos para ese mapplet.



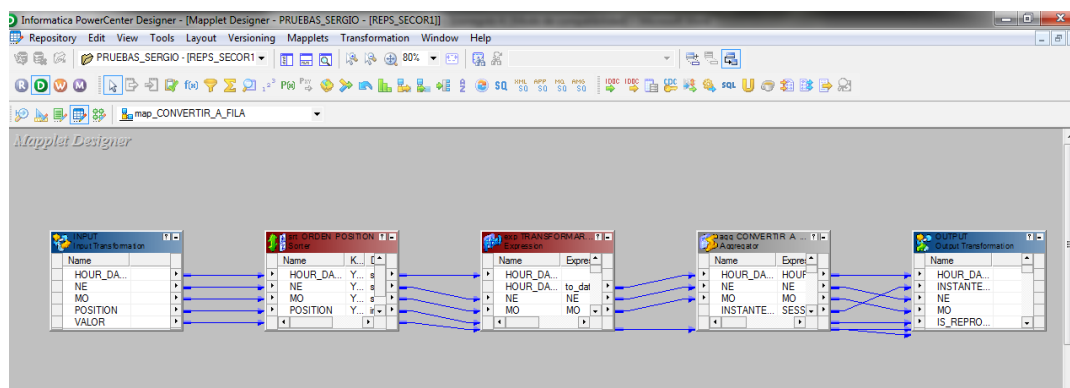
**Ilustración 93: Crear un Mapplet.**

Cuando pulsemos el botón ok se limpiara el espacio de trabajo de la vista de mapplets y podremos empezar a desarrollar la lógica. Para el desarrollo de estos mapplets al igual que para el desarrollo de los mappings que explicaremos en el siguiente apartado, estarán a disposición del desarrollador todos los componentes anteriormente mencionados como orígenes, destinos, procedimientos y funciones importados, transformaciones y una gran variedad de transformaciones predefinidas que ofrece la aplicación con el fin de poder desarrollar la lógica que podamos necesitar.



**Ilustración 94: Componentes disponibles para el desarrollador.**

El maplets completo tendría la siguiente forma:



**Ilustración 95: Ejemplo de maplet.**

Aquí vemos como se define un componente con las entradas que recibirá el maplet y después de realizar con esos datos las operaciones que sean necesarias para en este caso llegar a traspones una serie de registros de un solo campo en una única fila con varios campos colocaremos un nuevo componente que indicara la salida que saldrá como resultante de este maplet.

Una vez finalizado en maplet quedará disponible en nuestro menú de recursos que se muestra en la ilustración 94 en el apartado de maplet y este podrá ser usado en cualquier mapping a partir de ese momento.



### 2.6.2.5 Desarrollo de un mapping.

Es la base de los procesos de esta ETL, el mapping es quien desarrolla la mayor parte de la carga de los procesos ETL, ayudándose de los demás componentes creados en las vistas anteriormente explicadas y que junto con las otras tres aplicaciones que ya hemos comentado y de alguna de las cuales seguiremos comentando sus funcionalidades en paginas sucesivas del presente documento, realizan todos los procesos de carga necesarios en un proyecto que trabaje con esta ETL.

Los mapeos trazan el detalle de la lógica de carga que tendrá nuestro proceso de carga. Realiza todas las asignaciones y las transformaciones que sufrirán a lo largo de todo el proceso para llegar a obtener los datos tal y como los deseemos en nuestra fuente de destino.

En primer lugar para comenzar a desarrollar un mapping o mapeo (lo llamaremos así indistintamente), iremos al menú mapping de la vista de mappings anteriormente comentada y pulsaremos sobre la opción “Create”, en ese momento nos aparecerá una ventana para indicar el nombre del mapeo, la convención para nombrar los mapeos es que el nombre empiece con la letra “m” y continúe con guion bajo y el nombre que se desee poner, por ejemplo en nuestro caso se llamará “m\_PRUEBAS\_SERGIO”.

Una vez que hemos indicado el nombre y pulsado en ok aparecerá un espacio de trabajo en blanco en la vista de mappings. Ahora tendremos que comenzar eligiendo los orígenes de los cuales necesitamos recoger los datos que vayamos a cargar. Como hemos visto estos orígenes pueden ser una tabla de cualquier bbdd o ficheros de diversas extensiones. En nuestro caso serán unos ficheros delimitados por el signo “=” tal y como los que mostramos en los ejemplos anteriores.

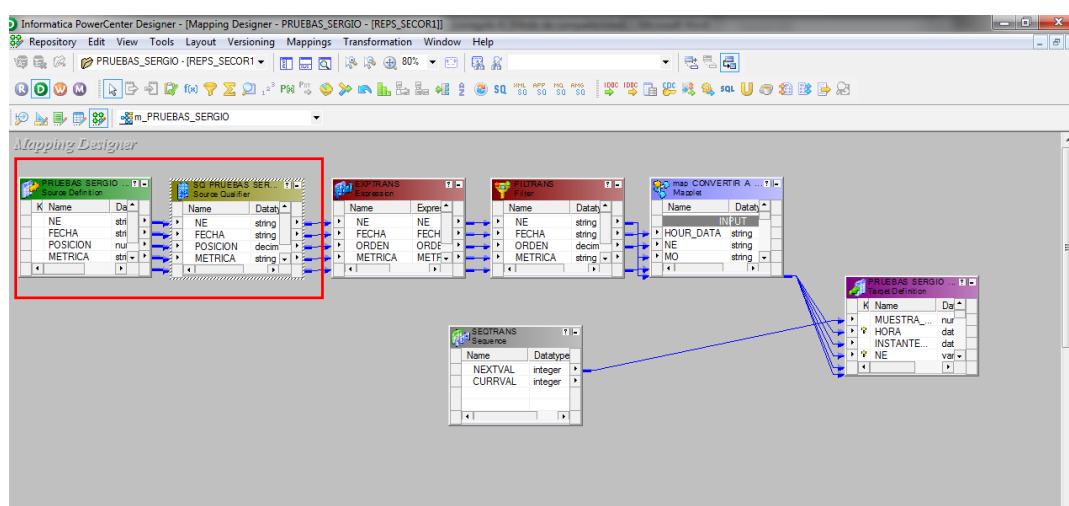


Ilustración 96: Desarrollo de un mapeo, orígenes.

Posteriormente a la realización de la lectura del origen, insertaremos un componente que se llama expresión y que es el componente principal y más usado en la aplicación ya que nos permite realizar diversos tipos de operaciones sobre cada uno de los campos. Para ello hay que entrar en el componente y en las opciones nos permite indicar que un campo será de salida y en ese momento nos permitirá editar la transformación que queramos hacer para ese campo. Los campos por defecto cuando se enlazan entre componentes son de entrada y salida, estos no sufren ninguna transformación en el paso de un componente al siguiente. Los componentes

de entrada, hay que indicarlos expresamente marcando ese check en la configuración del componente y en ese momento el dato de ese campo llegará a ese componente pero no saldrá del mismo. Y por último pueden ser solo campos de salida, estos suelen basarse en un campo de entrada o entrada-salida, se realizan las transformaciones u operaciones pertinentes sobre ese campo y se saca el resultado al siguiente componente. El último tipo de campo es el tipo variable, es un campo que no es de entrada ni de salida y que se utiliza como variables intermedias para transformaciones dentro del componente que estamos tratando.

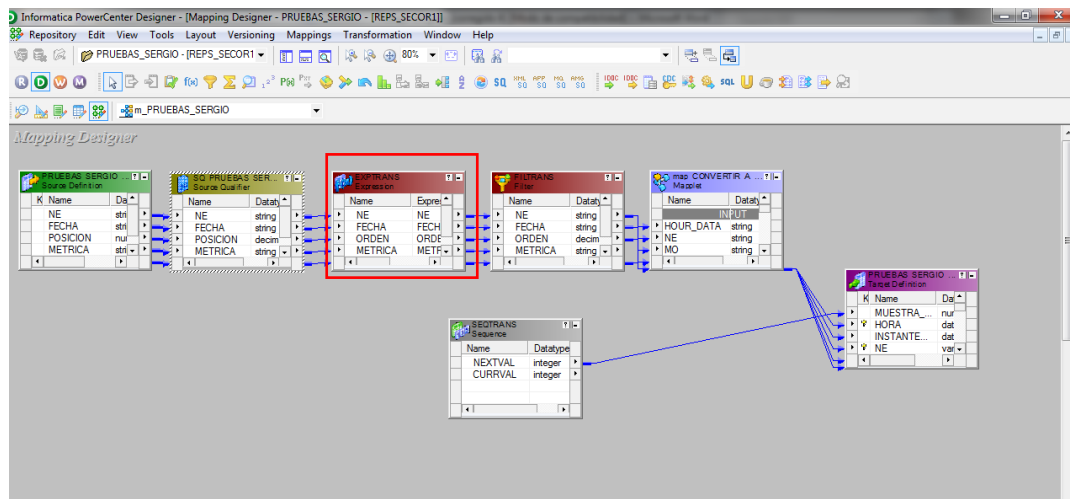


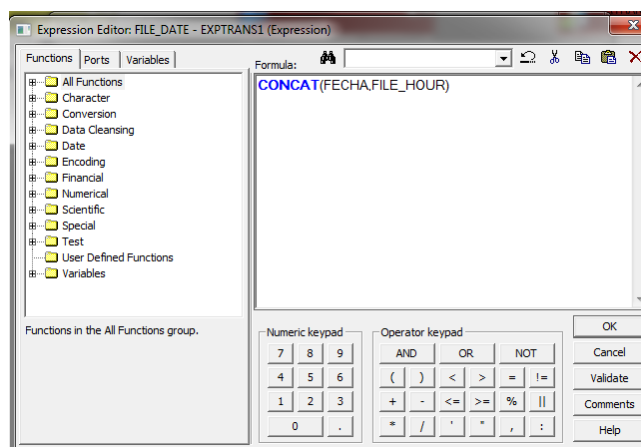
Ilustración 97: Desarrollo de un mapeo, expresiones.

En la siguiente ilustración se ven los tipos de campos que se pueden identificar, en este caso vemos los campos de entrada y salida que no pueden ser modificados. Después los campos de salida, este campo podría ser modificado realizando las operaciones que se muestran englobadas en la ilustración 99. También se ve el campo de entrada y por último la variable que al igual que los campos de salida puede verse modificada para realizar cualquier operación sobre el campo que se requiera, esta variable solo tiene validez o existencia dentro de este componente, no es una variable global que esas ya se explicaron en el apartado anterior [“Parámetros y variables”](#) como se creaban y que aquí también tendríamos disponibles para usarlas cuando nos fueran necesarias.

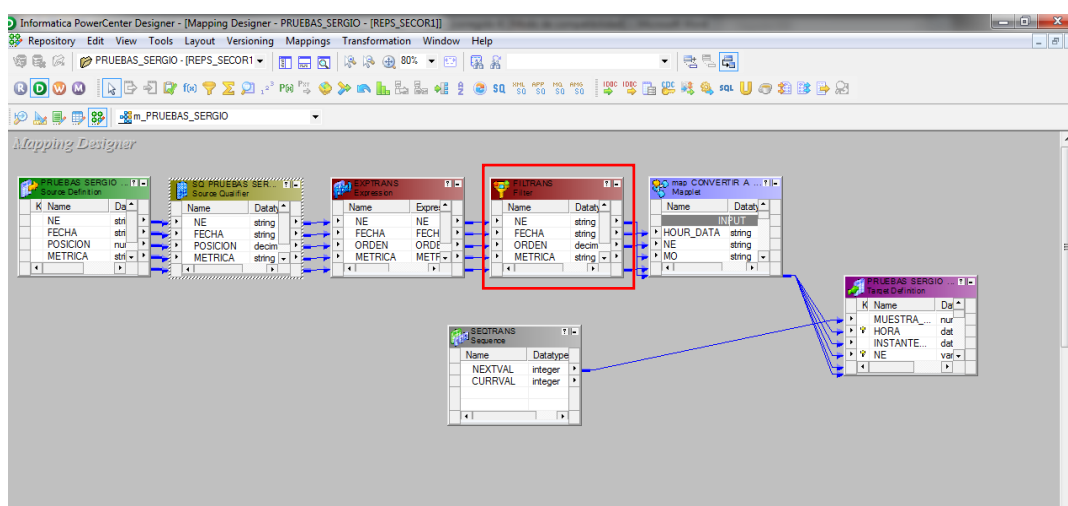
	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	NE	string	5	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NE
2	FECHA	string	19	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	FECHA
3	ORDEN	decimal	2	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ORDEN
4	METRICA	string	45	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	METRICA
5	VALOR	decimal	2	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VALOR
6	FECHA_OUT	date/time	19	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EXPRESIONES
7	NEWFIELD	string	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	EXPRESIONES
8	NEWFIELD1	string	10	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EXPRESIONES

Ilustración 98: Diseño de un mapeo, clasificación de campos en una expresión.

En la ilustración 99 podemos ver los grupos de funciones predefinidas que tiene la aplicación, aquí podemos modificar los campos según las necesidades que tenga el desarrollador, además también podríamos usar funciones definidas por el usuario y las variables locales y globales que se hayan definido.



**Ilustración 99: Diseño de un mapeo, operaciones sobre campos.**



**Ilustración 100: Diseño de un mapeo, filtros.**

En la ilustración 100 vemos como desde la expresión anteriormente comentada todos los campos que han entrado son los campos de salida de esa expresión, a su vez, estos son de nuevo entradas, que no han sufrido en la expresión ningún tipo de modificación, para un nuevo componente que ofrece las características de un filtro. Este nuevo componente realiza con que en bbdd sería la condición de un where, filtrando así los datos que recibe según las necesidades del proceso de carga.

De igual manera en los filtros se pueden definir campos de entrada y salida, campos que son de entrada pueden sernos útiles para realizar el filtro pero que no sea necesario que los saquemos para el siguiente componente ya que no los insertaremos finalmente en bbdd. En nuestro caso todos los campos son de entrada y salida tal como vimos en la expresión.

En la pestaña “properties” una vez que hemos entrado en la configuración del componente de filtrado aparecerá un recuadro ampliable con el nombre “Filter condition” en ese cuadro se

podrá indicar las condiciones de filtrado que se desean, si clickeas sobre la flecha que aparece en el recuadro cuando lo selecciones se abrirá una nueva ventana de edición igual que la de la [ilustración 99](#) y en la que podremos indicar el filtro que deseamos que se realice.

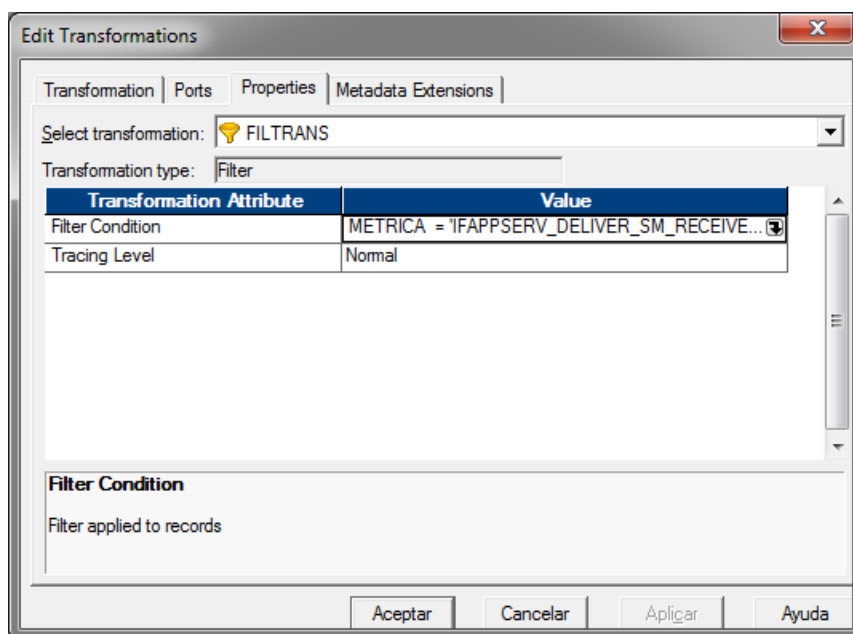


Ilustración 101: Diseño de un mapeo, filtros.

En este caso la condición que se indica es la siguiente:

```
"METRICA = 'IFAPPSERV_DELIVER_SM_RECEIVED' OR
METRICA = 'IFAPPSERV_DELIVER_SM_SENT_OK' OR
METRICA = 'IFAPPSERV_DELIVER_SM_SENT_KO' OR
METRICA = 'IFAPPSERV_DELIVER_SM_RSP_RECEIVED' OR
METRICA = 'IFAPPSERV_DELIVER_SM_RSP_TIMEOUT' OR
METRICA = 'IFAPPSERV_DELIVER_SM_RSP_SENT_OK' OR
METRICA = 'IFAPPSERV_DELIVER_SM_RSP_SENT_KO' OR
METRICA = 'IFAPPSERV_DATA_SM_RECEIVED' OR
METRICA = 'IFAPPSERV_DATA_SM_SENT_OK' OR
METRICA = 'IFAPPSERV_DATA_SM_SENT_KO' OR
METRICA = 'IFAPPSERV_DATA_SM_RSP_RECEIVED' OR
METRICA = 'IFAPPSERV_DATA_SM_RSP_TIMEOUT' OR
METRICA = 'IFAPPSERV_DATA_SM_RSP_SENT_OK' OR
METRICA = 'IFAPPSERV_DATA_SM_RSP_SENT_KO' "
```

Esta indica que si el campo métrica trae cualquiera de estos valores ese registro pasaría el filtro, de no traer estos valores indicados ese registro se desecharía.

Una vez que los registros son filtrados, los que han superado el filtro llegan como entrada a un mapplet (componente que hemos visto en apartados anteriormente explicados). Este realizará las transformaciones que tenga programadas para este mapping, en este caso llegarán varios registros que serán cada uno una columna de un nuevo registro, en otras palabras realiza una trasposición o cambio de varias filas con unos campos en común y un campo valor a un registro con esos campos comunes y varias columnas para distintos campos.

En la siguiente imagen representa la lógica que contendrá el mapplet que estamos explicando, en ella podemos ver como tenemos el componente de entrada, que simplemente refleja que campos son los que recibirá y con los que va a trabajar el resto de componentes del mapplet.

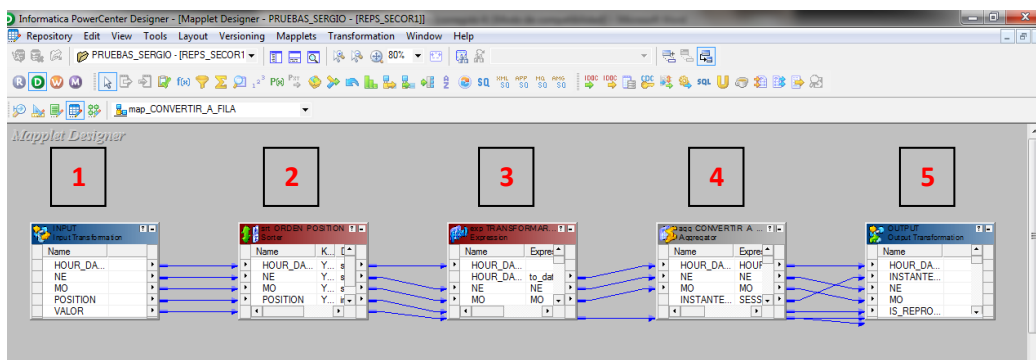


Ilustración 102: Diseño de un mapeo, lógica del mapplet.

Ahora podemos ver como una vez que recogemos los campos de entrada en el componente 1, se pasan directamente al segundo componente que realizará un ordenado de los registros según el/los campos que le indiquemos, en este caso cogeremos los cuatro primeros campos y los ordenaremos de forma ascendente, el campo posición indicará que posición de campo ocupará ese valor dentro de nuestro nuevo registro una vez que lo hayamos traspuesto.

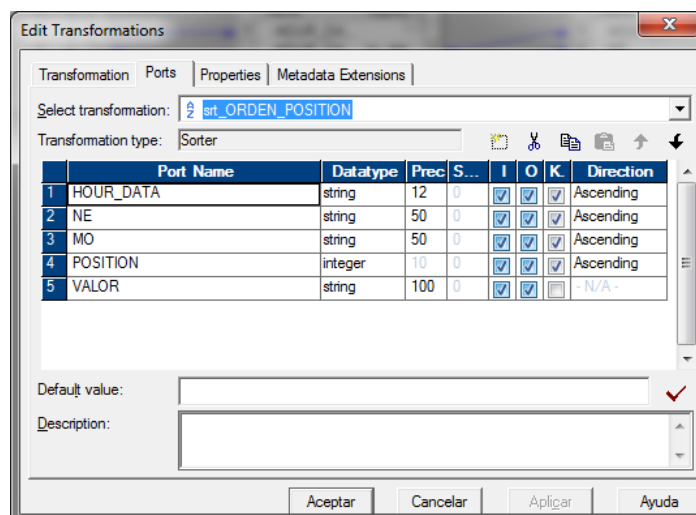


Ilustración 103: Diseño de un mapeo. Configuración de un componente de ordenación.

Una vez que hemos ordenado los datos estos pasan a una expresión intermedia que va generando el nuevo registro, manteniendo la misma clave va generando registros en los cuales dependiendo de que posición se indique que ocupa el valor generará un registro en el cual se mantendrá la clave y todos los campos de valor serán "null" salvo el campo que indique la posición en concreto que ocupa el valor que trae el registro (ejemplo en tabla 1). La configuración de este componente la podemos ver en la ilustración 104.

hour_data	NE	MO	POSITION	VALOR1	VALOR2	VALOR3
10/10/2013 12:00	NE1	MO1	1	1	NULL	NULL
10/10/2013 12:00	NE1	MO1	2	NULL	2	NULL
10/10/2013 12:00	NE1	MO1	3	NULL	NULL	3

Tabla 1: Representación de los valores de los registros antes de la agregación.



Una vez que hayamos agrupado y tengamos un único registro por clave llevaremos estos campos al componente de salida del maplet y este a su vez devolverá el flujo de ejecución al mapeo principal que estábamos viendo anteriormente con los resultados obtenidos de transformar los registros de entrada.

Finalmente, procederemos a insertar estos registros ya con el formato adecuado a nuestras necesidades en la tabla de destino, antes incluiremos un secuenciador que nos generará de manera automática los id para cada registro que insertemos en la tabla.

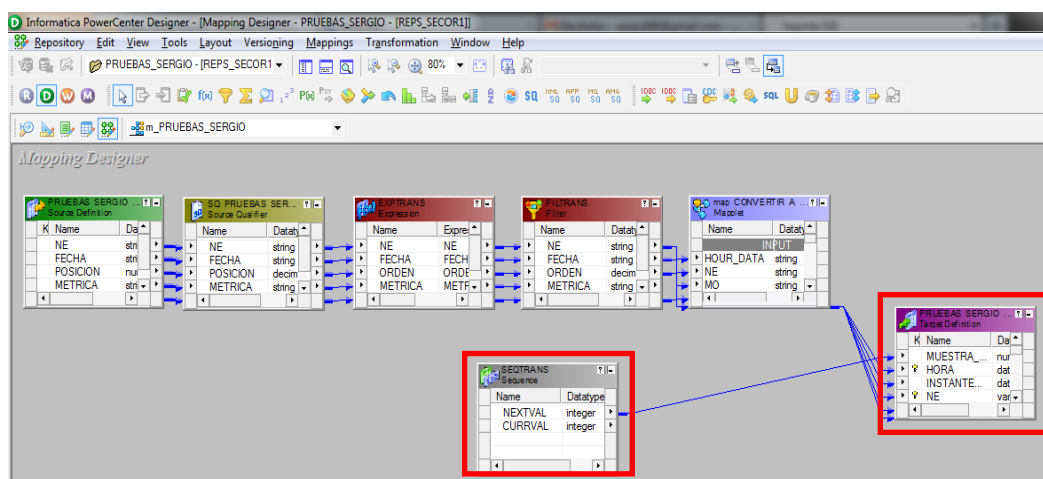


Ilustración 106: Diseño de un mapeo, secuenciador y destino.

## 2.6.3 Funcionalidades del workflow manager.

Esta aplicación desarrolla la lógica del proceso a un mas alto nivel, por realizar un símil con java, podríamos decir que lo que se ha visto hasta ahora con los mapeos serian los equivalentes a crear las clases que contienen la lógica a mas bajo nivel de la aplicación. Ahora con esta aplicación y con sus productos que se denominan workflows (abreviándolo wf) serán los equivalentes a crear las clases que contienen el main de java en la cual se instancian todos los objetos y se indican las líneas generales que va a seguir la aplicación.

Esta aplicación tiene como unidades para formar los workflows, unas estructuras denominadas sesiones que están basadas en los mapeos anteriormente realizados y que serian como las instancias de estos mapeos. Además tiene una serie de componentes propios al igual que tenia la anterior aplicación del designer y que complementaran y ayudaran a crear una lógica más compleja a nuestros procesos de carga.

Ahora vamos a explicar las funcionalidades más usadas de esta aplicación y finalizaremos con un ejemplo de workflow.

### 2.6.3.1 Unidades básicas.

En este apartado vamos a explicar las unidades básicas y mas comunes que ofrece esta aplicación para configurar la lógica a mas alto nivel de nuestros procesos de carga.

#### Sesiones.

Son componentes que implementan un mapeo que hayamos creado anteriormente en el mapping designer, estas sesiones instanciaran y darán valores reales en tiempo de ejecución a cada una de las entradas y parámetros que contenga el mapeo. Estos componentes son los más usados de toda la aplicación y son el pilar fundamental de los procesos de carga, pues son los que contienen la lógica de carga a más bajo nivel.

Para crear una sesión tenemos que ir a la vista task developer que tiene la aplicación workflow manager y o bien pulsar sobre el icono o bien pinchar sobre el menú “Task/Create...”

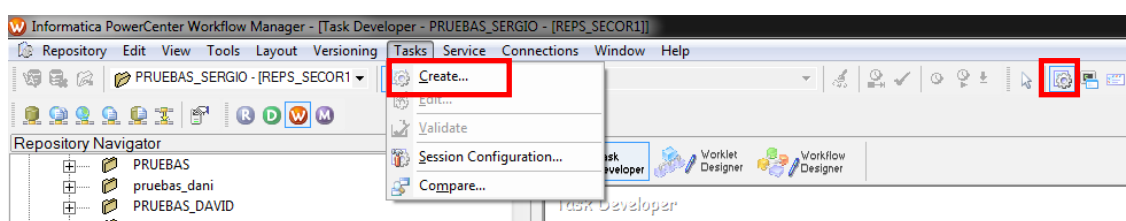


Ilustración 107: icono y menú de creación de sesiones.

Una vez pulsemos si pinchamos en el icono luego deberemos ir al espacio de trabajo de esa vista y pinchar sobre él para que nos cree el icono de la sesión, cuando lo hagamos, saldrá una ventana con todos los mapeos disponibles para ese directorio y tendremos que elegir uno en el cual se basará la sesión.

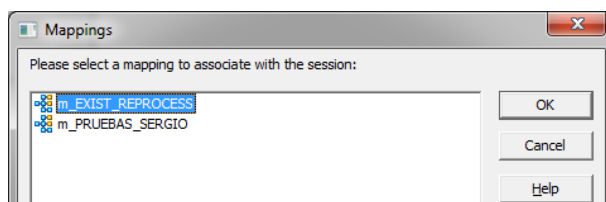


Ilustración 108: Ventana de selección de mapeo.

Una vez creado daremos doble clic para entrar en sus opciones de configuración.

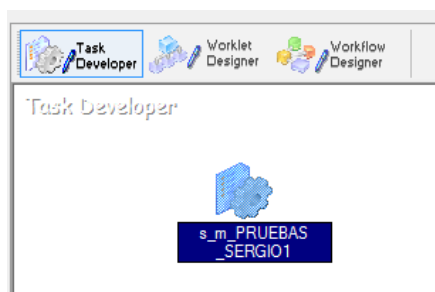


Ilustración 109: icono de sesión.

Cuando entremos en sus opciones de configuración aparecerá una ventana con distintas pestañas en la cual tendremos que configurar distintas opciones, resaltaremos las opciones más relevantes.



En la primera pestaña “general”, lo único que se puede realizar es el cambio de nombre que recibe la sesión, por defecto ésta cogerá el nombre que tenga el mapeo en el que se basa añadiendo el prefijo “s\_m\_nombre\_mapeo”.

En la segunda pestaña, “properties”, básicamente se configuran las opciones generales de la sesión y en este caso, configuramos donde queremos que nos deje los logs de ejecución de la sesión y el nombre que recibirá ese fichero.

Además podríamos asignarles a todos los orígenes y a todos los destinos unas conexiones por defecto de las que tengamos configuradas en el explorador de conexiones de la aplicación, estas conexiones podrán ser directamente a una bbdd o a través de un odbc.

De igual manera se puede indicar en esta pestaña la ubicación y el fichero que contendrá los parámetros que recibirá esta sesión o el intervalo de registros que debemos insertar antes de hacer el commit en bbdd.

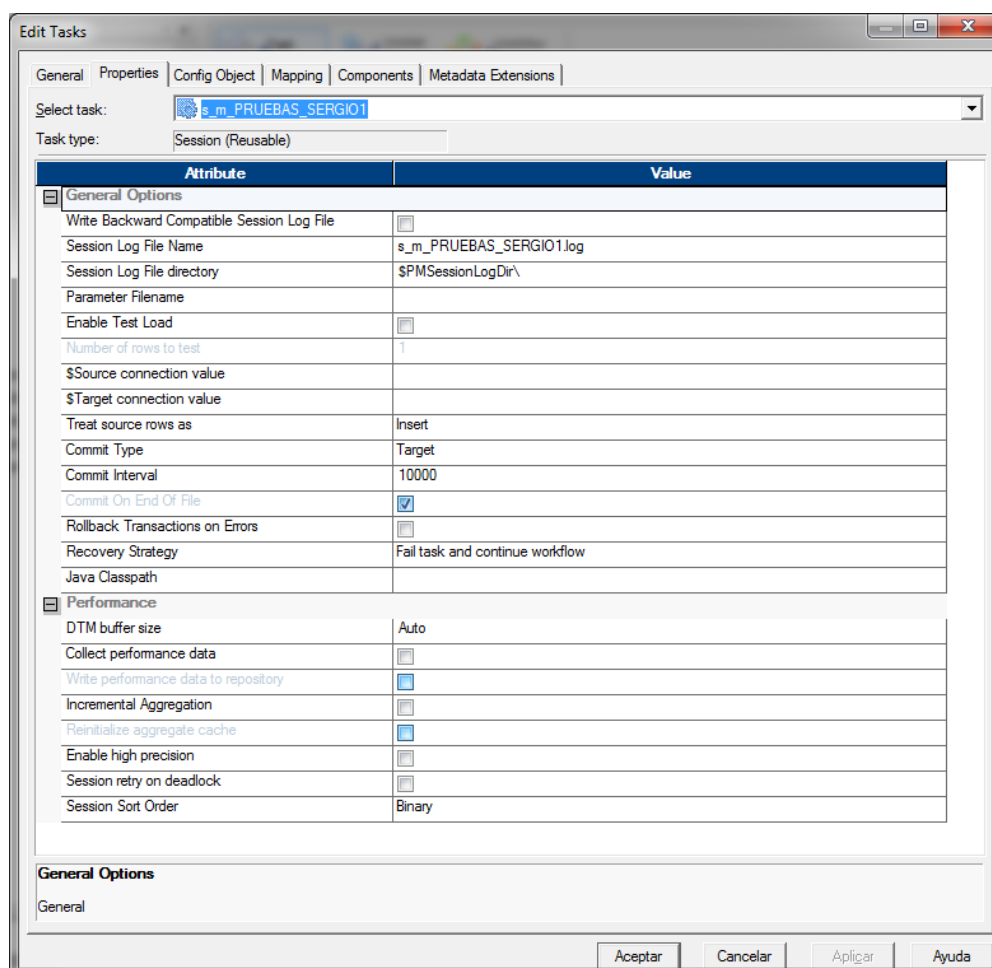


Ilustración 110: pestaña de propiedades.

En la siguiente pestaña, “config object”, podremos indicar las opciones de configuración a nivel “hardware” que tendrá esta sesión, por ejemplo podremos indicar cuanta memoria asignamos a la sesión, que haremos con el elemento lookup, si queremos que lo cachee en memoria o no, como queremos que guarde los logs si que guarde los logs de varias ejecuciones de sesiones o

que se sobrescriban los mismos en cada ejecución o las opciones de particionado si las fuentes o los destinos están particionados.

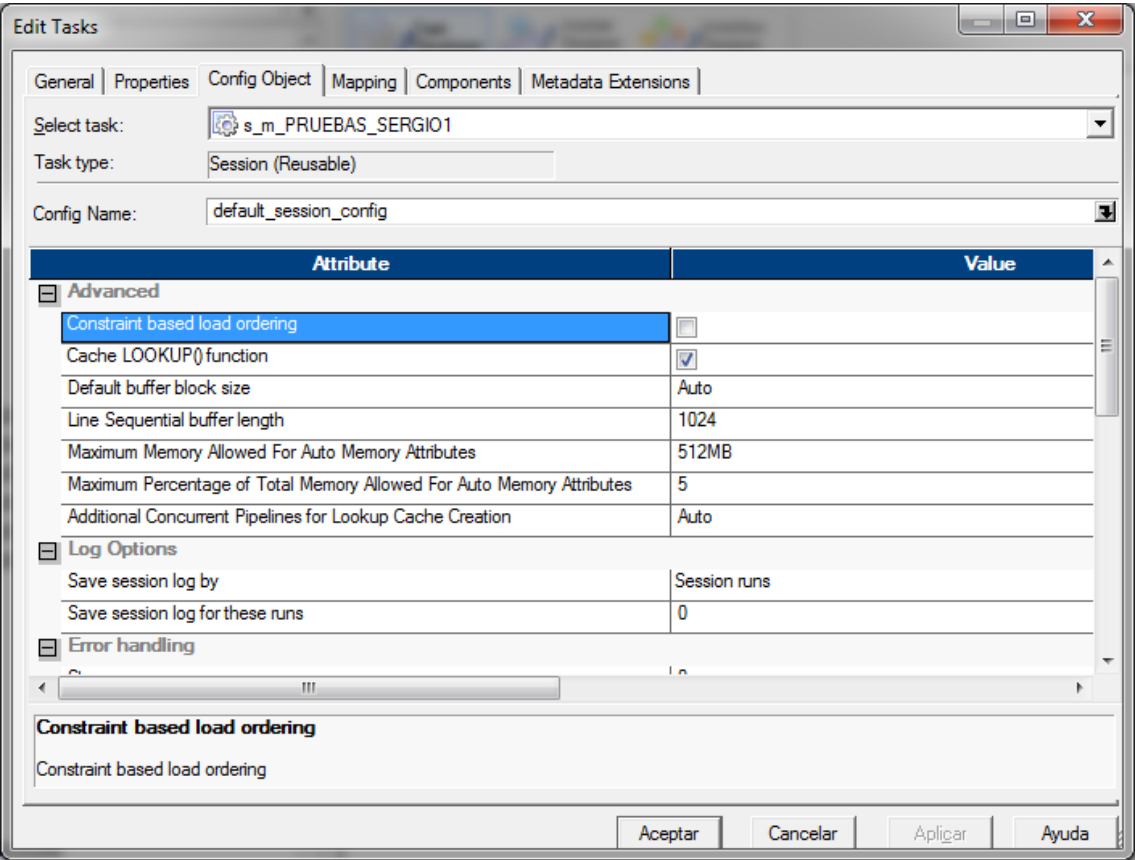


Ilustración 111: pestaña de opciones de configuración del objeto sesión.

Por último la pestaña “mapping”, es la que va a indicar a cada componente que forme el mapeo de donde va a tener que recoger o donde va a tener que dejar concretamente la información de la que disponga. En esta pestaña es donde se “instancia” cada uno de los componentes del mapeo.

La ventana estará dividida en dos secciones, la parte de la izquierda es un explorador de todos los elementos que contiene el mapeo y la derecha mostrará las opciones configurables para cada elemento seleccionado en el explorador de la izquierda. Normalmente se dejan todos los componentes con sus valores por defecto menos los orígenes y los destinos que si que hay configurarlos manualmente.

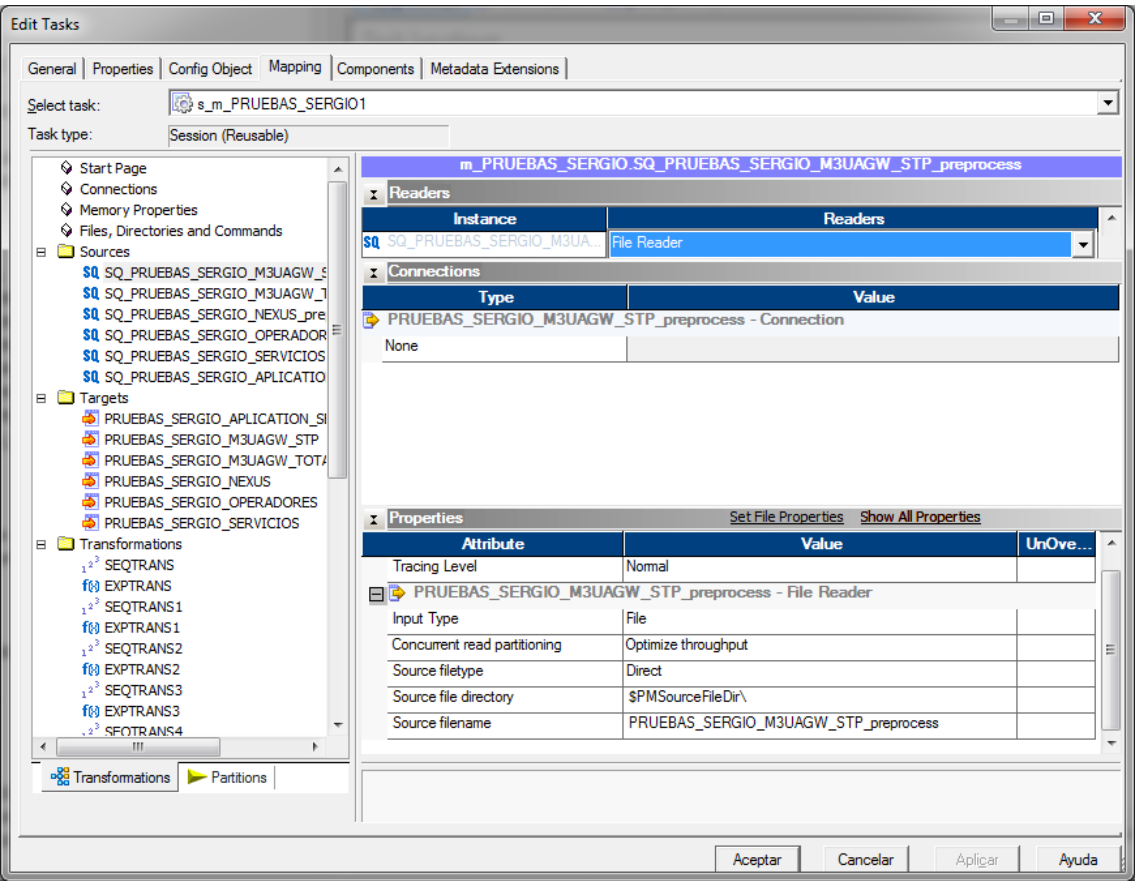


Ilustración 112: configuración de elementos de la sesión. Orígenes.

En la ilustración 112, vemos las opciones disponibles para configurar los orígenes, en primer lugar se debe indicar que tipo de componente de entrada es, pueden actuar como orígenes principalmente tablas de bbdd o ficheros. En caso de ser ficheros podremos elegir que tipo de lectura de fichero es entre “file reader” o “web services provider reader for flat file”.

Las opciones de conexión no aplican entre los orígenes de tipo fichero de texto pero si aplicarían entre las tablas de bbdd, esto mismo se explicará en los destinos ya que poseen las mismas opciones los orígenes que los destinos de bbdd y en este caso nuestros destinos serán bbdd.

En las propiedades del fichero podemos indicar que nivel de logs queremos que nos deje, que tipo de fichero es directo (contiene los datos a cargar directamente en el fichero) o indirecto

(contiene la ruta que apunta al fichero el cual contiene los datos que vamos a cargar). También puede indicarse en esta vista el directorio y el nombre del fichero que será nuestro origen.

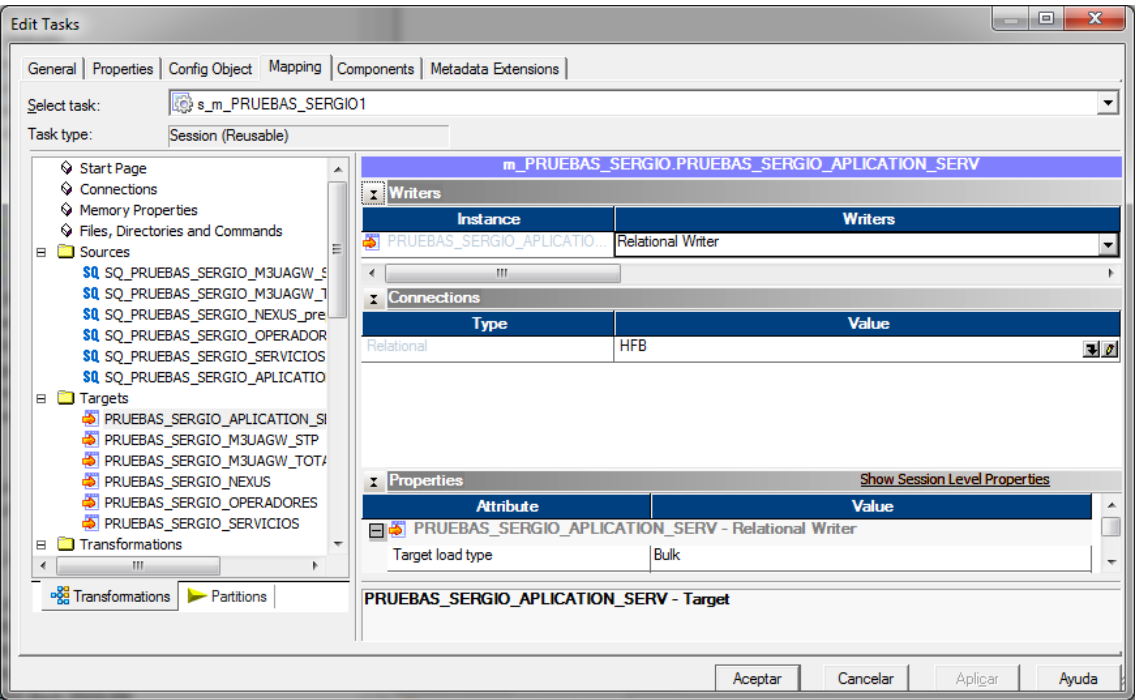


Ilustración 113: configuración de elementos de la sesión. Destinos.

En el apartado de configuración de los destinos podemos indicar el tipo de escritura que realizaremos entre un “relational writer” cuando escribamos en bbdd o “file writer” cuando escribamos en un fichero.

En caso de elegir una tabla como destino, se nos habilitará la opción de “connections” en la cual tendremos que indicarle a que bbdd nos vamos a conectar para escribir (o leer, para los orígenes de bbdd sería igual) de todas las conexiones que tenemos definidas en la aplicación.

En la ilustración 115 podemos ver las opciones de configuración para una tabla de bbdd, entre ellas podemos elegir que tipo de carga queremos realizar “bulk” realiza el commit al final de todas las inserciones que se vayan a realizar y “normal” realiza el commit cada x registros insertados. Además podemos indicar si el componente update\_strategy por defecto realizará insert, updates o deletes. También en este apartado se puede indicar si queremos que se trunque la tabla antes de insertar y los ficheros donde queramos que se escriban los logs de los registros rechazados.

Por último esta vista de opciones permite al programador indicarle que antes o después de que se ejecute la sesión lance un comando sql para lo que pueda resultar útil la información recogida con ese comando.

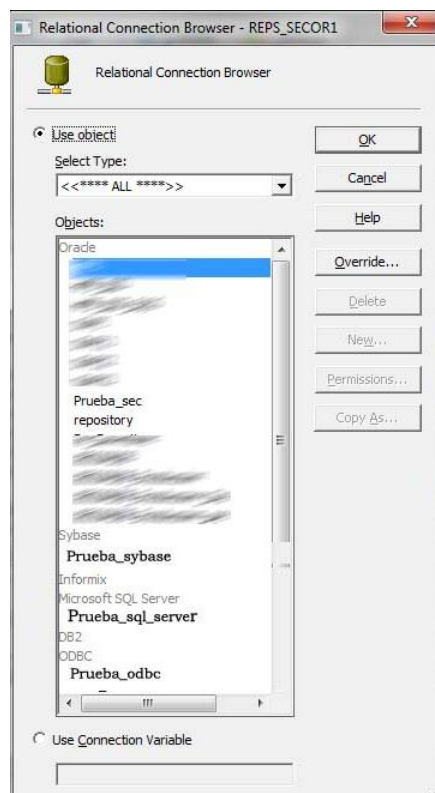


Ilustración 114: explorador de conexiones a BBDD.

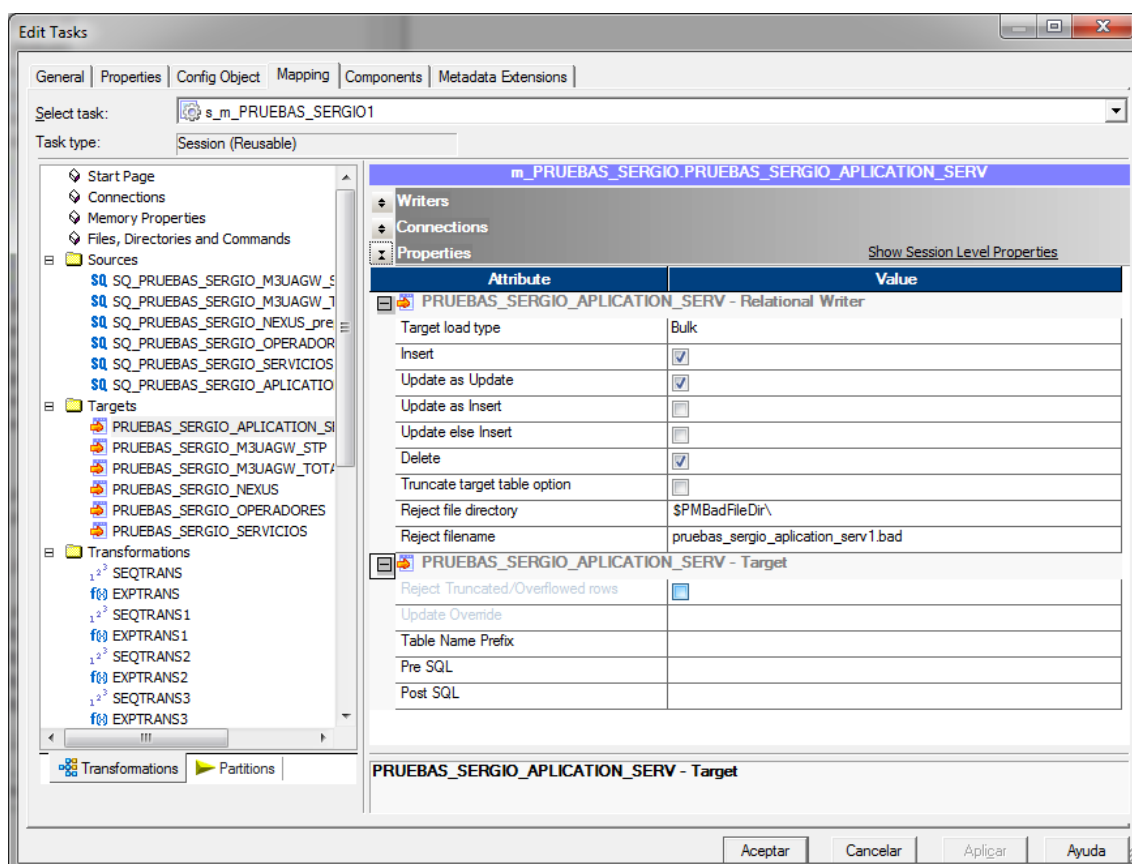


Ilustración 115: Propiedades de los destinos.

Para finalizar la configuración de las sesiones, en la pestaña de “Components” se puede elegir entre las opciones de lanzar comandos de consola o bien llamadas a scripts que realicen las tareas que se estimen oportunas, antes de ejecutar la sesión o una vez ejecutada que se lance ese comando si la sesión se ha ejecutado correctamente o si se ha ejecutado con algún fallo.

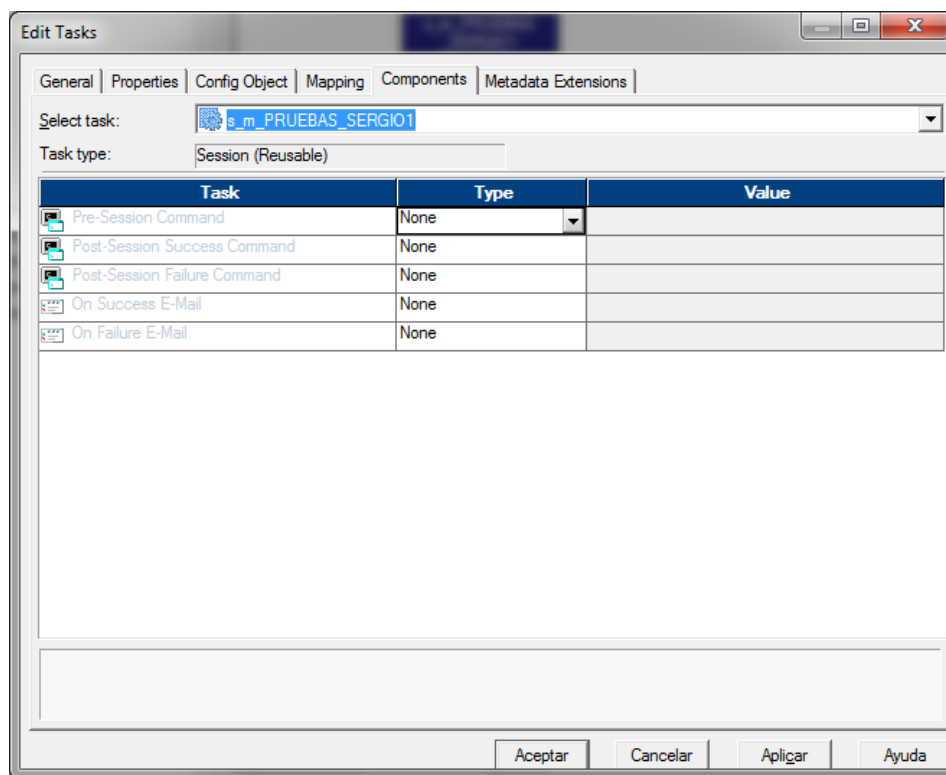



Ilustración 116: Pestaña Components.

### Commands.

Éste es el segundo componente mas utilizado te permite lanzar todos los comandos o scripts que necesites en tu proceso, pudiendo así encapsular lógica de carga y procesado en distintos scripts y por ello creando una lógica del programación tan compleja como necesites.

Para ello tienes que en la vista “Task developer” pulsar sobre el icono  o bien en el menú “Task/créate” seleccionar el tipo de tarea “command” y ponerle el nombre y después pulsar ok.

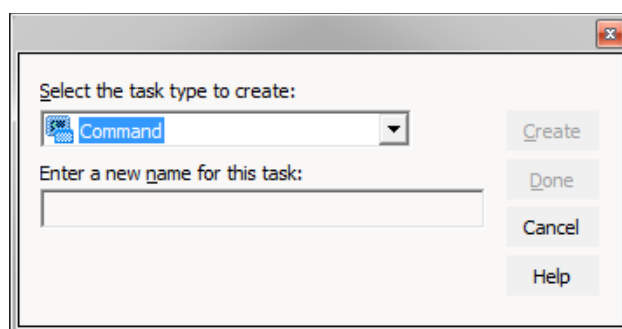


Ilustración 117: Crear un command.

Una vez creado tendremos que pulsar doble clic para acceder a sus opciones configuración y dentro de estas solo podremos configurar en la pestaña commands en la cual podremos crear tantas sentencias de comandos como necesitemos pulsando el icono que esta a la izquierda de la equis roja y después tendremos que pulsar sobre la flecha que sale en el campo en blanco command para que se nos abra otra ventana en la cual podremos indicar que comandos o scripts deseamos ejecutar.

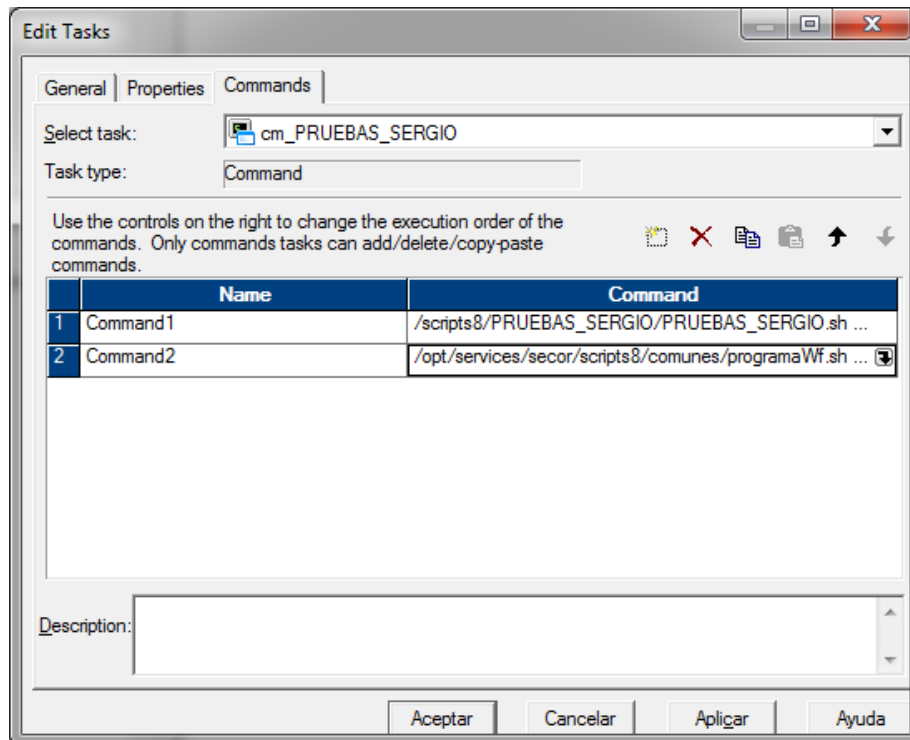


Ilustración 118: Ventana de configuración y creación de comandos.

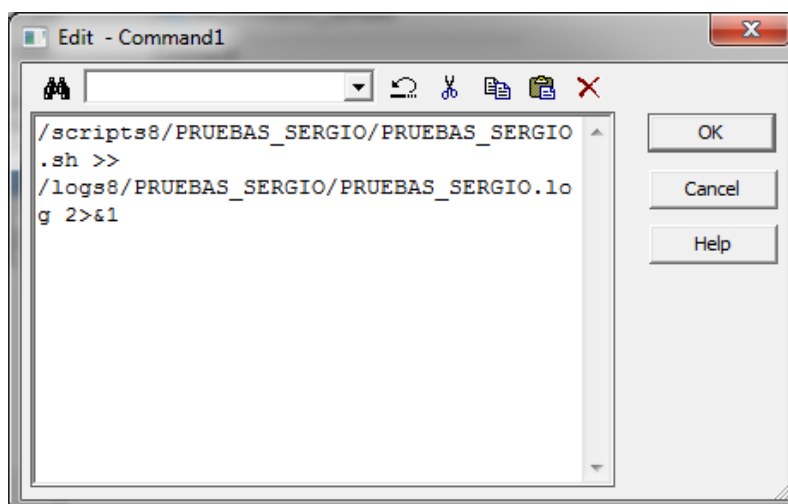


Ilustración 119: Ventana de edición de comandos.

### Otras unidades.


Otras unidades que se pueden utilizar son el componente email que te permite en un momento dado de la ejecución mandar un mail a los usuarios que desees y personalizar el texto que desees enviar.

El objeto “decisión” que permite bifurcar el flujo de carga atendiendo a alguna condición expresada.

Los objetos event wait o event raise, son objetos complementarios, si pones uno debes de poner obligatoriamente el otro. Estos lo que hacen es que uno espera que se produzca un evento definido por el desarrollador y el otro esta esperando la señal de que se haya producido el evento definido, en ese caso ejecutara el resto de sus componentes.

Los Worklets son agrupaciones de estos componentes que acabados de definir en este apartado y que al igual que ocurría con los mapplets encapsulan parte de la lógica y se pueden reutilizar en el flujo de carga del workflow principal.

#### 2.6.3.2 Programar un workflow

Para programar la ejecución periódica de un workflow, una vez creado el wf que queramos programar y que éste esté desplegado en la vista workflow designer iremos al icono  y una vez pulsemos sobre este icono se abrirá una ventana con las opciones de programación del workflow (wf), en dicha ventana tendremos que recurrir a la pestaña “Scheduler” para poder realizar la programación del mismo y pulsar sobre el icono que se muestra en la parte derecha para desplegar las opciones de programación.

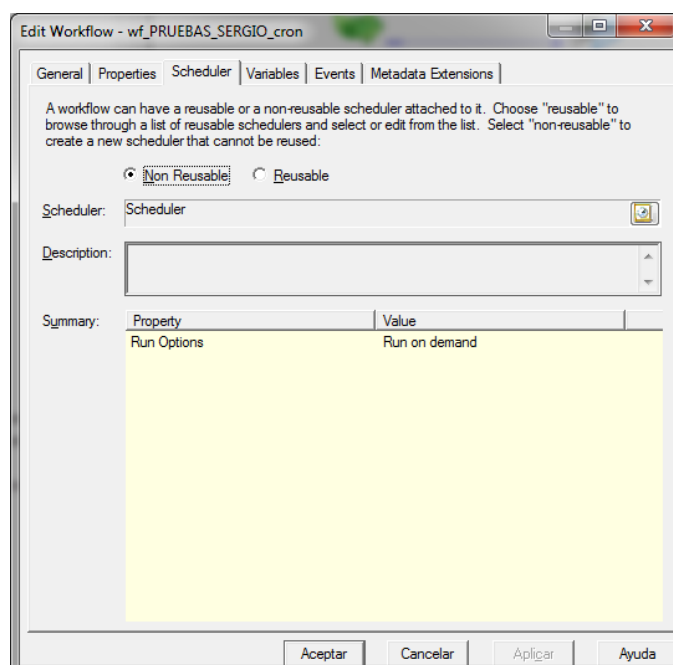
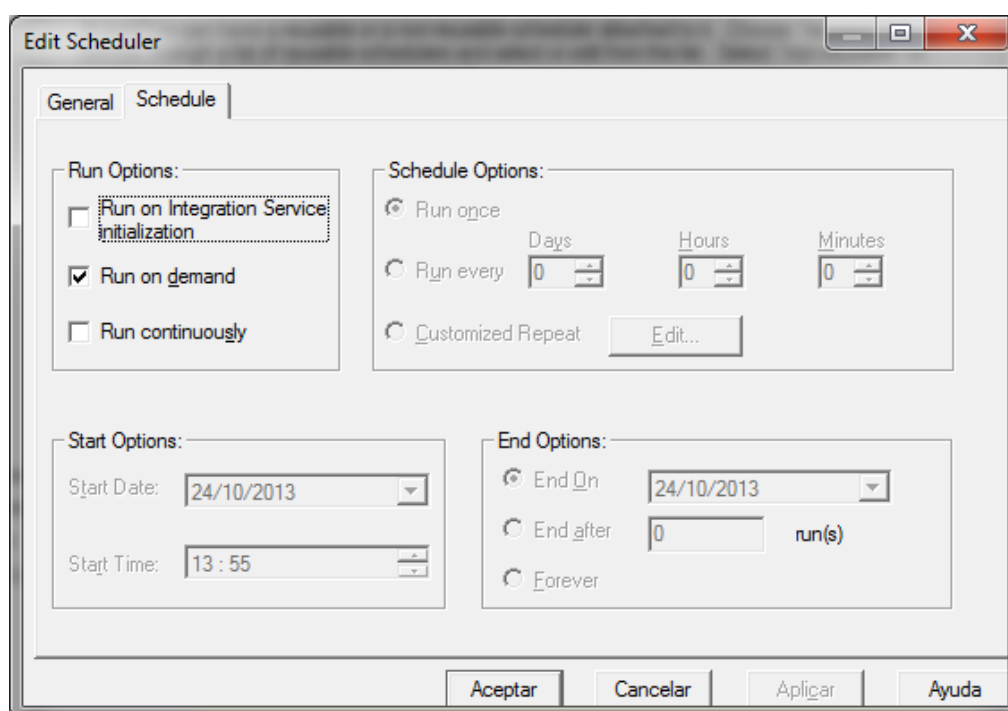


Ilustración 120: pestaña de programación.





**Ilustración 121: Ventana de programación de ejecuciones del wf.**

En la ventana de la ilustración 121 se ve la programación que tiene por defecto un workflow que es que éste se ejecute bajo demanda del programador. Si desmarcas esta opción se habilitan las otras tres secciones de la ventana y te permite configurar las horas de ejecución del wf.

En este caso en concreto vamos a indicarle que nuestro workflow se ejecute cada 1 hora, a partir del día 24/10/2013 a las 13:55 y que se ejecute siempre esta programación. Como podemos ver en la imagen se podría personalizar los intervalos de ejecución para poderlo programar y que no se ejecute los fines de semana, los festivos o cualquier otra restricción que necesitemos indicarle.

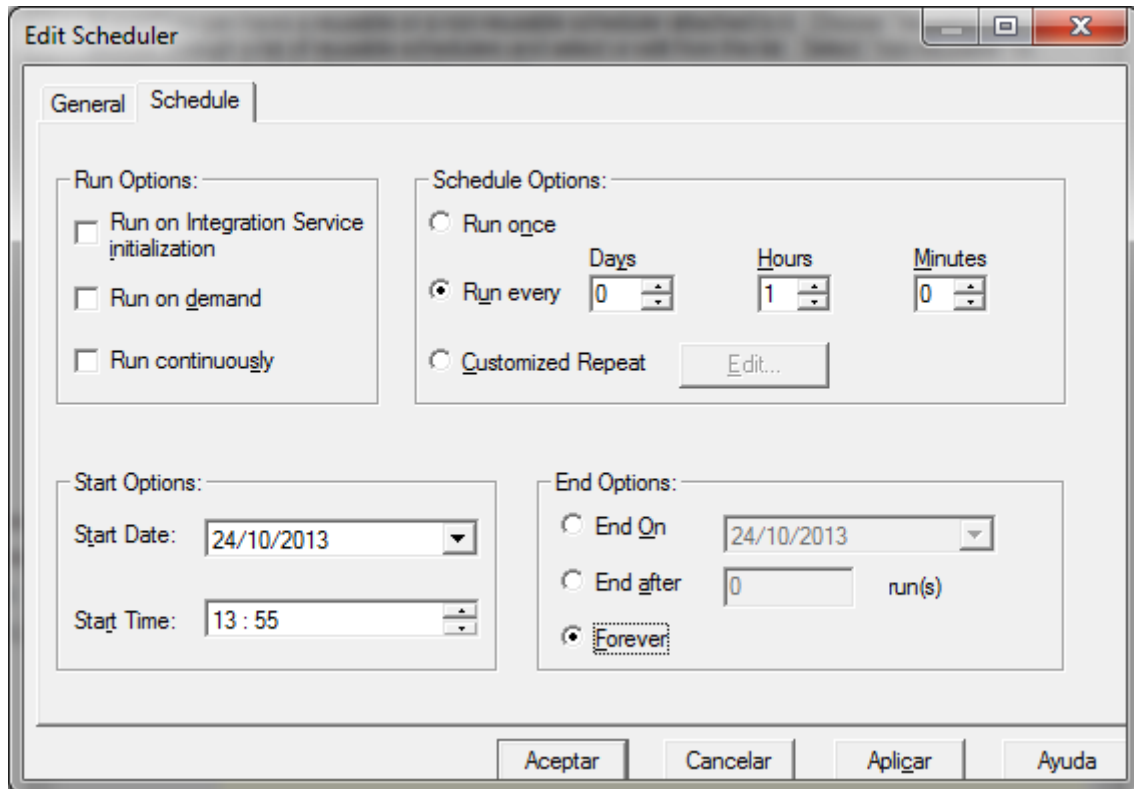


Ilustración 122: workflow programado para ejecución horaria.

### 2.6.3.3 Desarrollo paso a paso de un workflow.

En este apartado detallaremos la creación de un workflow (wf) paso a paso. Para ello en primer lugar tendremos que ir a la vista “workflow designer” y pulsar en el menú “workflows/Create” una vez pulsemos aparecerá una ventana en la cual nos saldrán varias pestañas en la primera tendremos que indicar el nombre que queramos dar al wf. El resto de pestañas se dejan con la configuración por defecto y solo al finalizar utilizaremos la pestaña Scheduler para programar el wf tal y como hemos visto en el punto anterior.

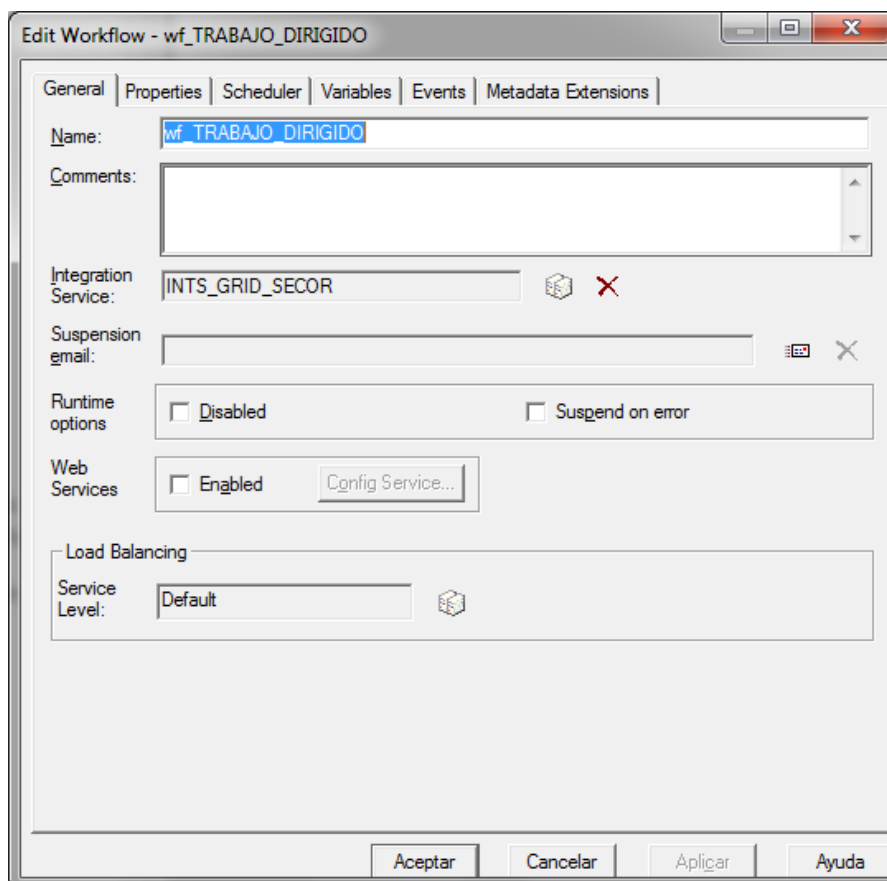


Ilustración 123: Ventana de creación y configuración de workflow.

Una vez pulsemos aceptar aparecerá en la vista de trabajo de workflow designer un icono de una flecha verde con el nombre start. A partir de ese icono se irán enlazando todos los componentes que formen nuestro workflow de carga.

En nuestro caso lo primero que añadiremos será una sesión basada en un mapping que recoge datos de un fichero y los inserta en tablas de bbdd. Esta sesión esta basada en el mapping que se creo en el punto [3.3.2.5 Desarrollo de un mapping](#). La sesión ya la habremos creado y configurado previamente tal cual se ha detallado en el apartado anterior de este presente documento. De esta manera, las sesiones creadas serán reutilizables en cualquier workflow, aunque es posible que una vez la utilices en el workflow la puedas editar sin que esa configuración específica para ese workflow modifique de alguna manera la configuración que le dimos cuando creamos la sesión reutilizable.

En la parte izquierda de la vista aparecerán todos los componentes que hemos creado anteriormente y que podrán ser utilizados en el desarrollo de la lógica del workflow.

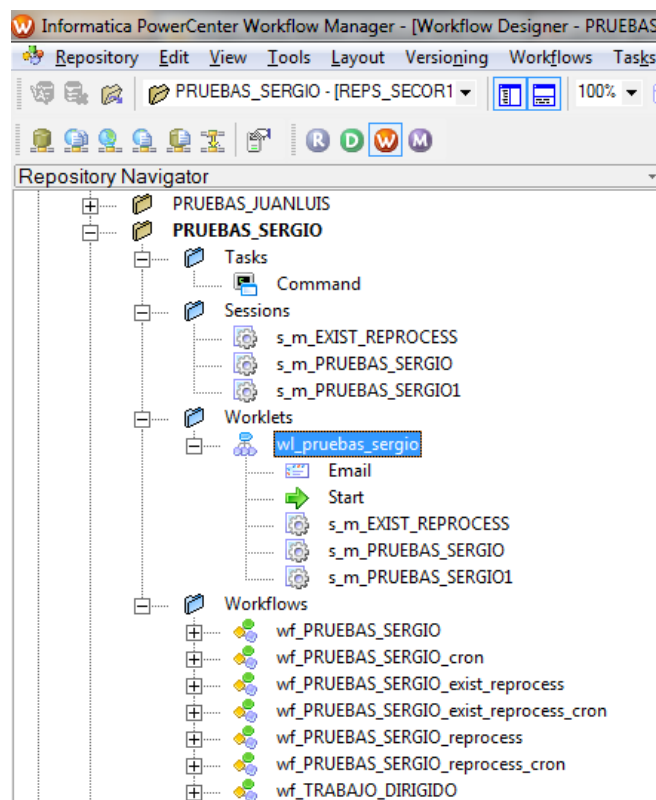


Ilustración 124: componentes utilizables en el workflow.

Las sesiones hay que enlazarlas mediante unas líneas de conexión que indicaren el flujo de ejecución.

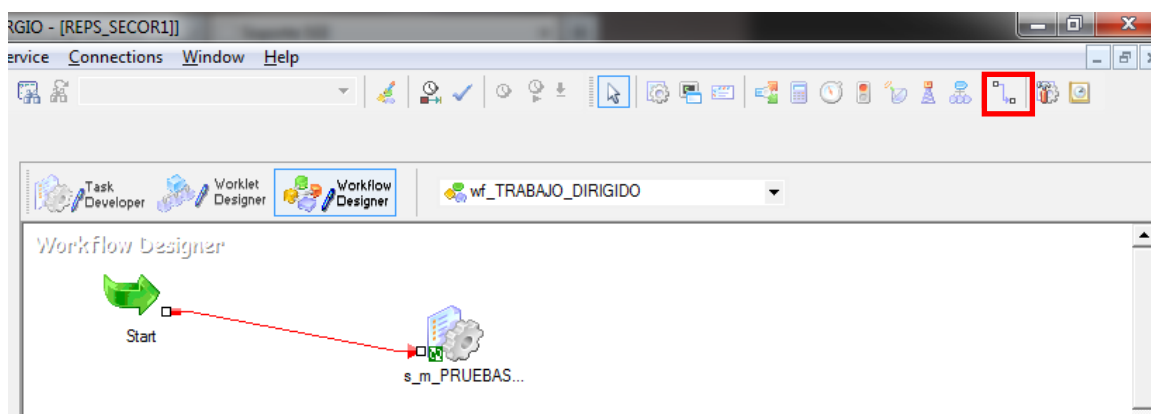


Ilustración 125: Estructura general del workflow, inserción de la primera sesión.

En la ilustración 124 podemos ver como hemos insertado en nuestro wf la primera sesión que se ejecutará y como la hemos conectado con la línea de conexión que se crear en el icono que aparece resaltado.

Estas líneas de conexión pueden ser editadas para modificar el comportamiento del workflow o para indicarle filtros o restricciones.

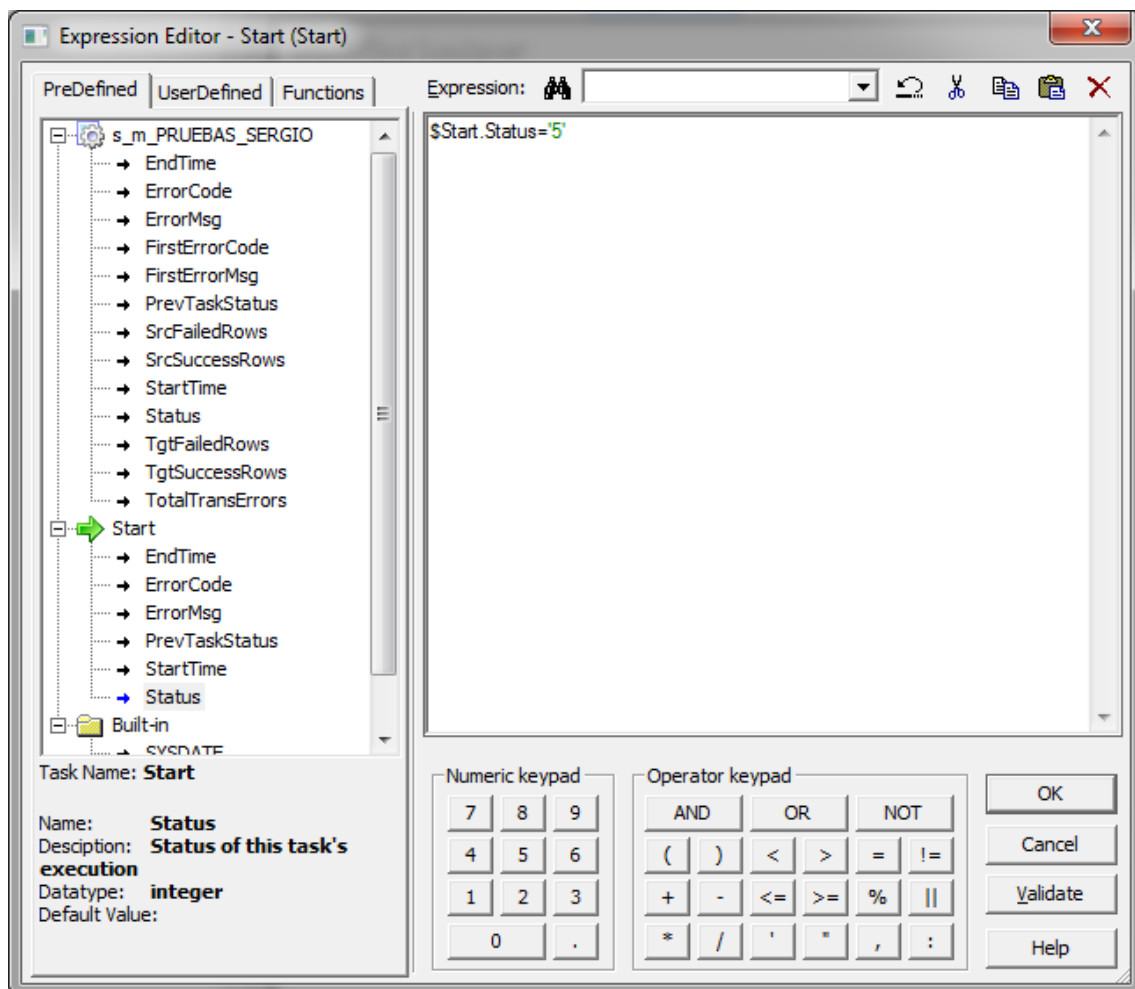


Ilustración 126: filtros en las líneas de conexión.

Una vez insertada la primera sesión vamos a añadir un worklet que contendrá otras dos sesiones y que al finalizar mandará un mail.

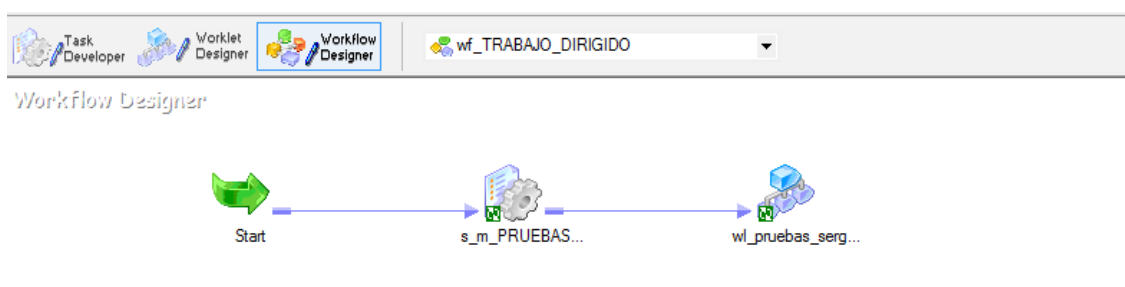
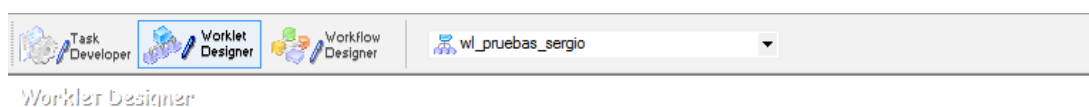


Ilustración 127: Estructura general del workflow, worklet.

Si pulsamos con el botón contractual del ratón sobre el worklet y pulsamos la opción “open worklet” se abrirá la vista de “worklet designer” y nos mostrará el contenido de éste.



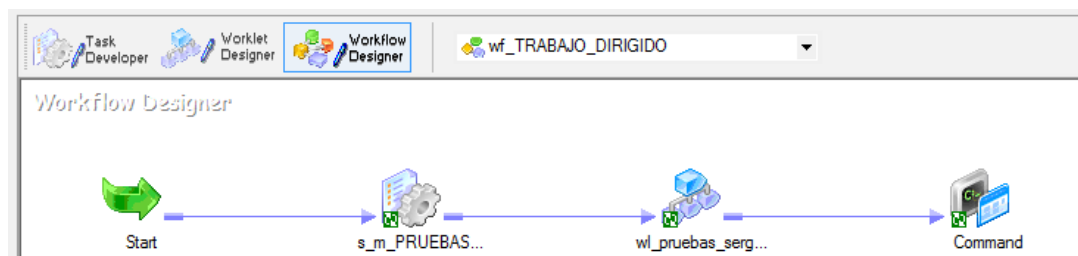
**Ilustración 128: diseño del workflow, contenido del worklet.**

En esta ilustración podemos ver como nuestro worklet contiene encapsulado dos sesiones basadas en dos mapeos distintos en la cual la primera se encarga de ver si alguna ejecución por algún motivo no se ha lanzado y se han metido reprocesos por falta de datos, comprobar que esos datos ya estén disponibles.

La segunda sesión realiza la carga de los datos en caso de que la primera sesión haya detectado que había disponibles datos listos para ser reprocesados.

Por último el último componente, manda un mail a un usuario indicándole que se van a reprocesar esos datos.

Finalmente de nuevo en la vista de workflow designer introducimos el último componente de nuestro ejemplo que es un “command,” creado en el apartado anterior y que lo que hará será lanzar un script que realizará distintas tareas y dejar los logs de éste en un directorio en concreto.



**Ilustración 129: Estructura general del workflow, inserción del command.**

Por último, una vez que ya tenemos toda la lógica del workflow desarrollada, realizaremos su programación tal y como mostramos en el apartado anteriormente detallado.

## 2.6.4 Funcionalidades del workflow monitor.

Esta herramienta permite monitorizar las ejecuciones de los procesos de carga así como revisar sus logs y realizar una programación y desprogramación de los workflows que ya tienen configuradas sus programaciones previamente desde el workflow manager (indicando periodos, horas y situaciones especiales que se comentaron en el apartado [3.3.3.2 Programar un workflow](#) del presente documento).

Como se ha indicado en el apartado de interfaz esta aplicación muestra la ejecución de los workflows con dos vistas un diagrama de gant y un listado con las tareas programadas y como va su ejecución.

Esta aplicación tiene la funcionalidad de mostrar al usuario toda la información perteneciente a los procesos que se están ejecutando y permite a su vez, reiniciar, parar, programar o desprogramar los procesos que ya estén programados.

### 2.6.4.1 Opciones sobre un workflow.

En la siguiente imagen podemos ver como el workflow esta programado, esto se observa si nos fijamos en la columna estatus (puede tener varios estados entre ellos scheduled, unscheduled, running, succeded, failed, abort u stopped) . Pulsando sobre él con el botón derecho del ratón podemos ver todas las opciones que tenemos, para inicializar este workflow tendremos que pulsar la opción “Restart”.

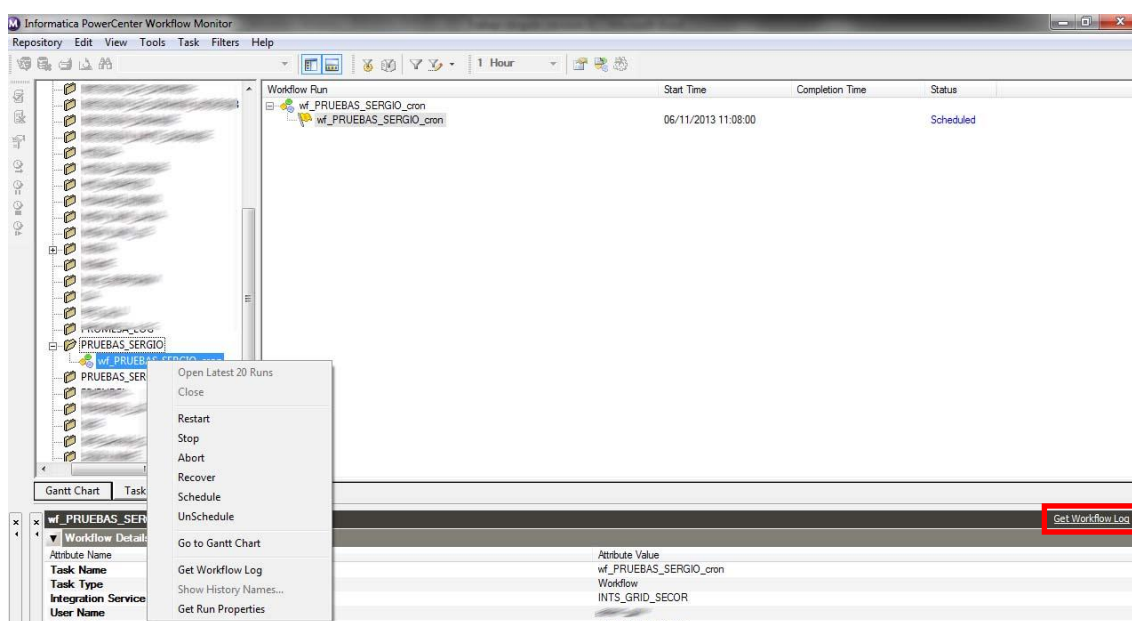


Ilustración 130: Opciones sobre un workflow.

En el momento en el que se pulse esta opción veremos como el estado cambia al estado “running”, lo que querrá decir que el proceso ya se esta ejecutando.

De igual manera podremos realizar las anteriormente citadas operaciones sobre los workflows que podrían ser: Parar la ejecución del workflow con la opción “Stop”, Abortar la ejecución con la opción “Abort” (la diferencia entre estas dos es que una espera a que finalice algunos procesos pendientes y la otra corta abruptamente la ejecución), también podemos programar de nuevo o desprogramar las ejecuciones con las opciones Schedule o UnSchedule.

### 2.6.4.2 Obtención de logs.

Desde esta aplicación podremos obtener dos tipos de logs, por un lado los logs de sesión y por otro lado los logs del workflow.

En principio, los logs que arroja la sesión son mas específicos y nos mostraran mas detalle de los posibles errores que el logs de los workflows, pero puede darse el caso de que el fallo se produzca en el paso de los parámetros o en el intento de conexión a la BBDD y que salte en los logs del workflow ya que no habría llegado a ejecutarse la sesión.

Para obtener los logs del workflow, tenemos que seleccionar un workflow y después podemos pulsar con el botón derecho del ratón sobre la opción “GetWorkflowLog” o si le seleccionamos podremos pulsar en la vista inferior sobre un link en el que también pone dicha opción. Esto lo podemos apreciar resaltado en la ilustración 130 mostrara anteriormente.

Una vez pulsada cualquiera de estas opciones aparecerá una nueva ventana que contendrá los logs correspondientes a la ejecución seleccionada.

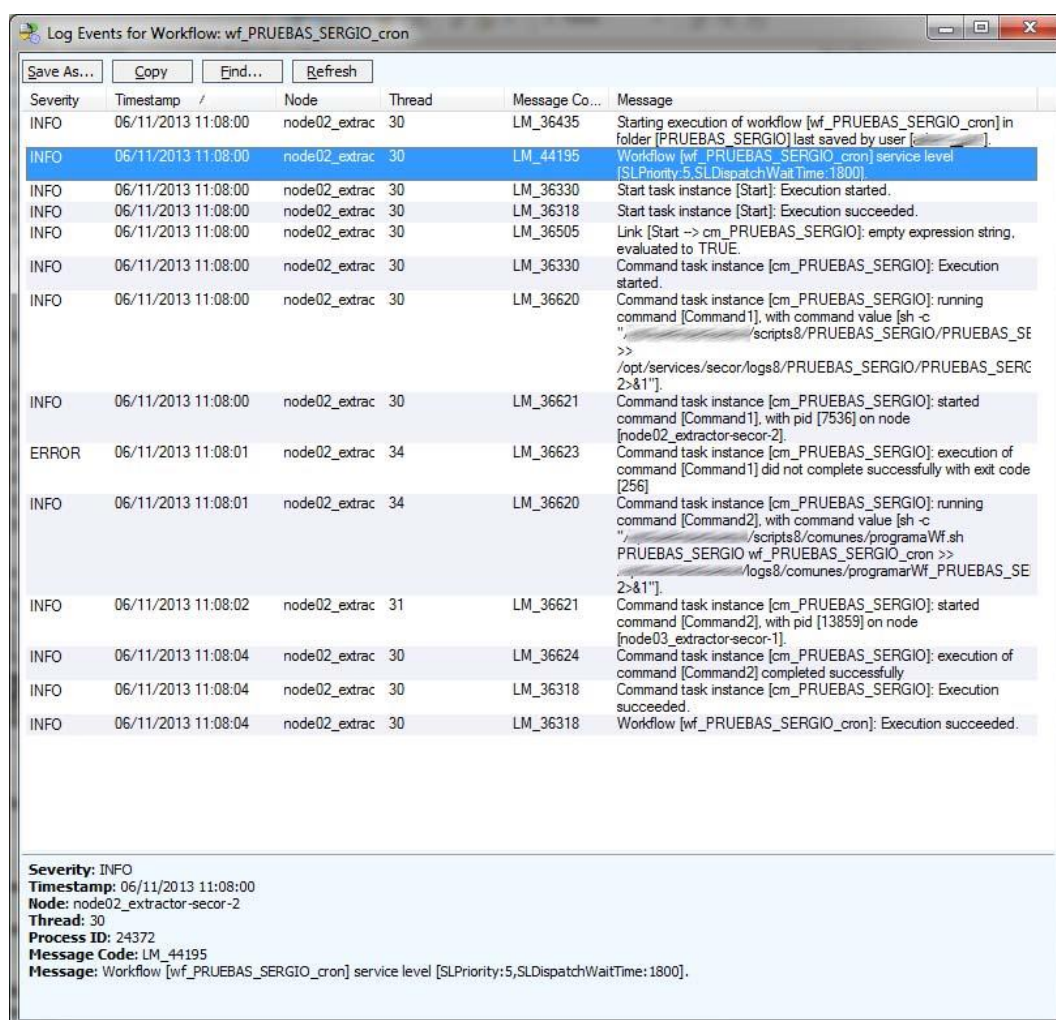


Ilustración 131: workflow log.

En esta ventana tendremos las opciones de buscar alguna palabra dentro del log, salvar el log completo, copiar un registro concreto o si lo abrimos durante una ejecución ir refrescando la



información cargada. Vemos igualmente que cuando seleccionamos un registro del log, la información sale completamente detallada en la parte inferior de la ventana.

Por último para obtener los logs de la sesión se realizarían los mismos pasos que para los de los workflows pero seleccionando la sesión de la cual queremos obtener los logs.

### 3 Prueba de concepto.

En este apartado vamos a ilustrar con unos ejemplos prácticos el desarrollo de distintas cargas de datos, procesando ficheros con varios formatos y realizando cargas similares con distintas estrategias y algoritmos de resolución para un mismo problema.

Aquí intentaremos transmitir al lector de este documento la potencia y el uso que se le puede dar a una herramienta como la anteriormente expuesta y cuál es su uso en fines profesionales.

Para realizar nuestra prueba de concepto vamos a tratar el siguiente ámbito de la vida real.

#### 3.1 Dominio.

Este estudio va a tratar de obtener distintas conclusiones acerca de los flujos de natalidad y mortalidad en España durante los últimos 10 años a nivel de poblaciones y municipios, así como ver de qué manera afecta a estas variables el índice de paro o las rentas por sector de cada una de las provincias o municipios.

#### 3.2 Modelado multidimensional de los datos.

Nuestro diseño de datos se compondrá de ocho tablas maestras que serán referenciadas mediante foreignkeys por nuestras tablas de hechos componiendo de esta manera un modelo multidimensional como se representa en el siguiente modelo de datos.

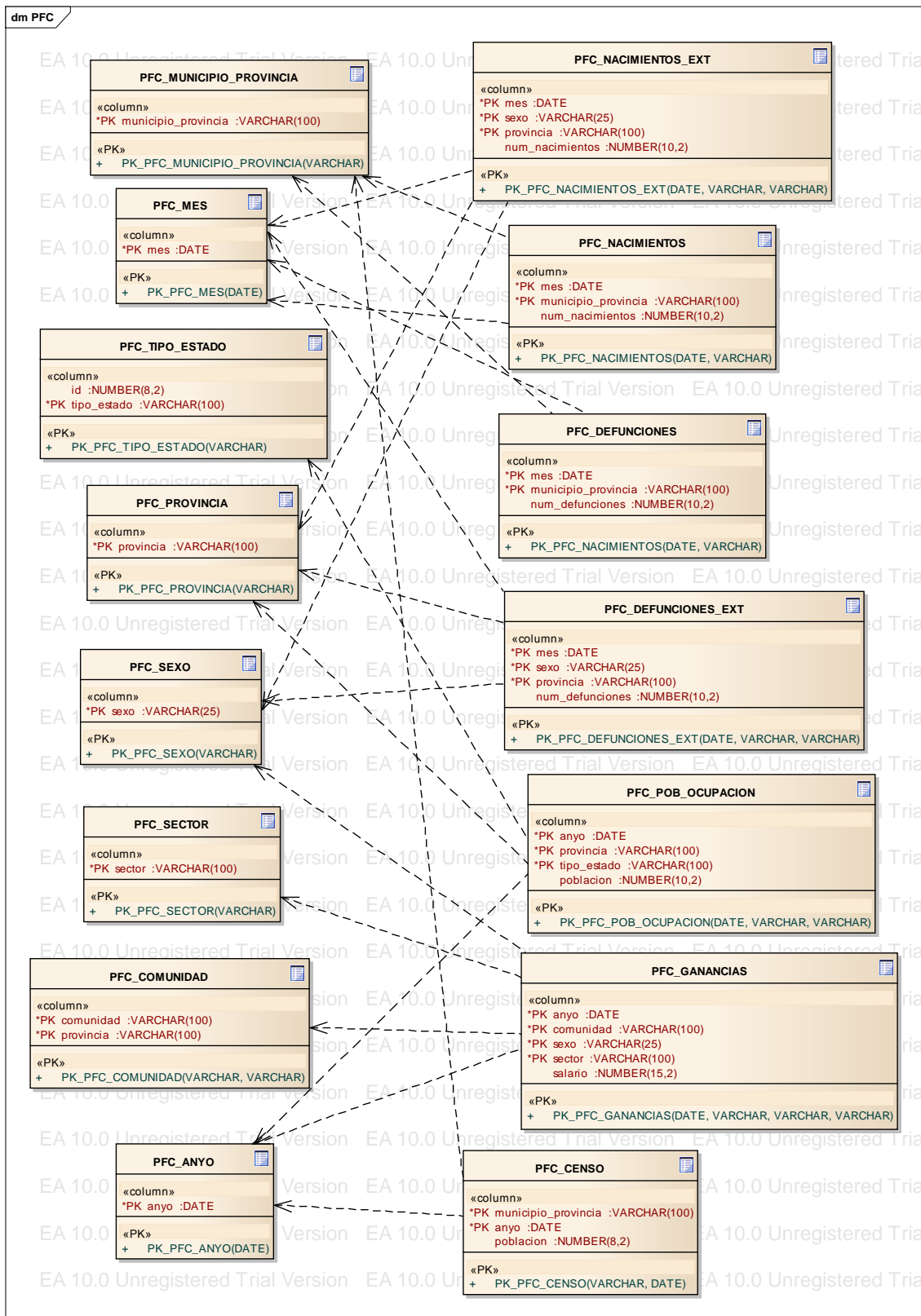


Ilustración 132: Modelo de datos.

A continuación os indicaremos la estructura de las tablas que componen nuestro modelo e indicaremos las dimensiones y posibles valores que tienen los campos multivaluados.

Campo	Tipo
Mes	date

Tabla 3: PFC\_MES.

Esta tabla será referenciada por todas las tablas de hechos cuyos datos estén insertados a nivel de mes. El formato que se insertará desde el proceso de carga será “Mes-AA”, por ejemplo “Enero-12”. Si bien en bbdd se guardará como un tipo de datos fecha lo que significará que se guardará con el formato que tenga definido el editor de sql que utilicemos y dicha fecha será siempre establecida a día 1 del mes correspondiente y del año indicado “01/01/2012”, teniendo que darle si se desea el formato que queramos al realizar la query sobre la tabla de hechos.

La tabla PFC\_SEXO contiene un campo sexo que tendrá como posibles valores “Hombre” y “Mujer”, valores que actúan como claves ajenas en todas las tablas de hechos en los cuales se realice una diferenciación entre sexos.

Campo	Tipo
Sexo	Varchar (25)

Tabla 4: PFC\_SEXO.

Otra de las tablas maestras que hemos definido es la concerniente al municipio y la provincia de la que ese registro es objeto de estudio. Los valores son un campo compuesto por cada una de las provincias concatenados con el símbolo “|” y el tamaño de los municipios, teniendo estos cuatro valores posibles (menos de 10000 habitantes, entre 10001 y 50000 hab., más de 50000 hab. y capital).

Campo	Tipo
Municipio_provincia	Varchar(100)

Tabla 5: PFC\_MUNICIPIO\_PROVINCIA.

En algunas comparativas hemos tenido que realizar el estudio a nivel de provincia ya que no existían datos de esas variables a nivel de municipio. Para ello hemos creado una tabla con los nombres de todas las provincias de España.

Campo	Tipo
Provincia	Varchar (100)

Tabla 6: PFC\_PROVINCIA.

Para el estudio realizado sobre la encuesta de población y su estado laboral, hemos precisado de otra tabla denominada PFC\_ESTADO, en la que se indica en qué estado se encuentra la persona en concreto, pudiendo ser estos estados: activo\_ocupado, activo\_parado, activo\_parado1empleo o inactivo. Además hemos necesitado incluir esta tabla un identificador que será usado en el algoritmo de carga de la tabla de hechos que estudia esta variable.

Campo	Tipo
-------	------

Tipo_estado	Varchar (100)
id	number

Tabla 7: PFC\_TIPO\_ESTADO.

En el estudio de los salarios anuales por provincia y sector hemos requerido del uso de otra tabla maestra que es PFC\_SECTOR, esta tabla contiene tres posibles valores que son: Industria, construcción y servicios.

Campo	Tipo
Sector	Varchar (100)

Tabla 8: PFC\_SECTOR.

Otra de las tablas maestras que tenemos es la tabla de años, esta tabla es utilizada por los procesos de carga cuando las muestras están recogidas en periodos anuales. Tiene un único campo numérico que contiene el año.

Campo	Tipo
anyo	Number

Tabla 9: PFC\_ANYO.

La última tabla maestra con la que contamos es con la tabla de comunidades. Algunos de los ficheros de muestra que tenemos están clasificados por comunidades en lugar de por provincias, de ahí que hayamos tenido que crear una tabla para poder cruzar los datos con dicha tabla.

Campo	Tipo
Comunidad	Varchar (100)
Provincia	Varchar (100)

Tabla 10: PFC\_COMUNIDAD.

Una vez descritas las tablas maestras utilizadas por los procesos de carga como claves ajenas, procederemos a explicar las tablas de hechos de las que disponemos.

En primer lugar disponemos de la tabla de nacimientos, llamada PFC\_NACIMIENTOS, esta tabla dispone de los campos: mes, municipio\_provincia y num\_nacimientos.

Campo	Tipo
Mes	Date
Municipio_provincia	Varchar (100)
Num_nacimientos	number

Tabla 11: PFC\_NACIMIENTOS.

De igual estructura es la tabla PFC\_DEFUNCIONES, tabla que contiene el número de defunciones nacionales a nivel de municipio-provincia por año.

Campo	Tipo
Mes	Date
Municipio_provincia	Varchar (100)

Num_defunciones	number
-----------------	--------

Tabla 12: PFC\_DEFUNCIONES.

Estas dos tablas se cargan mediante un mismo mapeo que se denomina m\_pfc\_nac\_def.

Otras dos tablas de hechos que se cargan simultáneamente son las que relacionan los nacimientos y las defunciones de personas extranjeras por mes, sexo y provincia de la defunción.

La estructura de ambas tablas es:

Campo	Tipo
Mes	Date
sexo	Varchar (25)
provincia	Varchar (100)
Num_defunciones	number

Tabla 13: PFC\_DEFUNCIONES\_EXT.

Campo	Tipo
Mes	Date
sexo	Varchar (25)
provincia	Varchar (100)
Num_nacimientos	number

Tabla 14: PFC\_NACIMIENTOS\_EXT.

De igual modo tenemos la tabla de censos, PFC\_CENSO, que nos permite ver a lo largo de los años como ha ido creciendo la población censada en cada municipio.

La estructura que tiene dicha tabla es la siguiente:

Campo	Tipo
Anyo	Date
Municipio_provincia	Varchar (100)
población	number

Tabla 15: PFC\_CENSO.

Una de las últimas tablas de hechos que tenemos es la de ganancias, PFC\_GANANCIAS, que permite comparar el nivel de ingresos comparando las variables del tiempo en años, la comunidad donde se trabaje, el sexo de la persona y el sector al que se dedique.

Campo	Tipo
Anyo	Date
Comunidad	Varchar (100)
sexo	Varchar (25)
Sector	Varchar (25)
salario	number

Tabla 16: PFC\_GANANCIAS.

En último lugar la tabla que cargaremos será la que estudia la ocupación de las personas en España a nivel provincial. Esta tabla estudia las variables según el año, provincia y el tipo de estado laboral en el que se encuentra la población estudiada.

Campo	Tipo
Anyo	Date
Provincia	Varchar (100)
Tipo_estado	Varchar (25)
población	number

Tabla 17: PFC\_POB\_OCUPACION.

### 3.3 Estructura de las fuentes.

Las fuentes de origen de datos que hemos utilizado para estos procesos de carga son ficheros obtenidos de la web del Instituto Nacional de Estadística (<http://www.ine.es/>).

Estos ficheros están con un formato en .csv separados mediante puntos y comas, cada fichero dispone de unas líneas de cabecera que explican que son las variables a estudiar y las propias cabeceras de los campos. Dichas líneas se pre procesan en algunos casos mediante un proceso de Shell script de unix previo y en otros casos se procesan desde el propio Powercenter. Si bien, todos los ficheros o gran parte de ellos en su “cuerpo” o parte de datos de carga han tenido que ser pre procesados de alguna manera para que coincidieran las claves ya que varios ficheros que representaban la misma información de varios años sufrían cambios en algunas nomenclaturas, por ejemplo en los nombres de las provincias, variaban su formato.

A continuación explicaremos cada fichero como lo recibimos en bruto de la web y como queda tras su pre procesado para su posterior carga mediante Powercenter.

Los ficheros de nacimientos y defunciones nacionales tienen el siguiente formato previamente a ser procesados:

Nacimientos.Datos por provincias y capitales

Nacimientos. Año 2003

Nacimientos por provincia de residencia de la madre, tamaño del municipio y capital de residencia de la madre y mes del nacimiento.

Unidades:nacimientos

;Todos los

meses;Enero;Febrero;Marzo;Abril;Mayo;Junio;Julio;Agosto;Septiembre;Octubre;Noviembre;Diciembre;

Total;"";"";"";"";"";"";"";"";"";"";"";"";"";"";

Provincia;"441881.0";"36189.0";"33051.0";"36853.0";"36944.0";"38593.0";"36100.0";"37849.0";"36814.0";"38655.0";"37978.0";"36117.0";"36738.0";

Capital;"146111.0";"11959.0";"11080.0";"12182.0";"12325.0";"13019.0";"11995.0";"12445.0";"11932.0";"12648.0";"12598.0";"11797.0";"12131.0";

Municipios de hasta 10.000

habitantes.Total;"85432.0";"7062.0";"6238.0";"7197.0";"7271.0";"7374.0";"7058.0";"7364.0";"7230.0";"7556.0";"7141.0";"6912.0";"7029.0";

Municipios de 10.001 a 50.000

habit.Total;"118375.0";"9667.0";"8906.0";"9839.0";"9777.0";"10276.0";"9591.0";"10106.0";"9999.0";"10385.0";"10242.0";"9739.0";"9848.0";

Municipios > 50.000

habitantes.Total;"91963.0";"7501.0";"6827.0";"7635.0";"7571.0";"7924.0";"7456.0";"7934.0";"7653.0";"8066.0";"7997.0";"7669.0";"7730.0";

Álava;"";"";"";"";"";"";"";"";"";"";"";"";"";"";

Provincia;"2693.0";"198.0";"187.0";"212.0";"259.0";"278.0";"232.0";"232.0";"193.0";"238.0";"219.0";"210.0";"235.0";





habitantes.Total;"1908.0";"195.0";"143.0";"145.0";"163.0";"152.0";"165.0";"163.0";"159.0";"163.0";"166.0";"158.0";"136.0";  
[...]

Fuente: Instituto Nacional de Estadística

Copyright INE 2014

Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 -  
Contacta: [www.ine.es/infoine](http://www.ine.es/infoine)

**Tabla 18: Formato de ficheros de nacimientos y defunciones nacionales.**

Como se puede apreciar, estos ficheros contienen unas líneas iniciales con la cabecera y la explicación de que datos se presentan en ellos y finalmente tienen un pie de fichero con distintos datos sobre las fuentes.

Son ficheros separados por “;” y de un año a otro podemos encontrar variaciones con respecto al nombre o formato del nombre empleado en el fichero.

Una vez pre procesados los ficheros tienen el siguiente formato:

```
DEFUNCIONES;2003;Defunciones.Datos por provincias y capitales
DEFUNCIONES;2003; Defunciones. Año 2003
DEFUNCIONES;2003;
DEFUNCIONES;2003;Defunciones por provincia de residencia tamaño de municipios y capital y
mes de la defunción.
DEFUNCIONES;2003;Unidades:defunciones
DEFUNCIONES;2003;
DEFUNCIONES;2003;
DEFUNCIONES;2003;;Todos los
meses;Enero;Febrero;Marzo;Abril;Mayo;Junio;Julio;Agosto;Septiembre;Octubre;Noviembre;Di
ciembre;
DEFUNCIONES;2003;Total;"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "",
DEFUNCIONES;2003;
Provincia;"384828.0";"35814.0";"31863.0";"32431.0";"30596.0";"30332.0";"31816.0";"31288.
0";"34729.0";"27569.0";"29
859.0";"32370.0";"36161.0";
DEFUNCIONES;2003;
Capital;"130836.0";"12171.0";"10654.0";"11104.0";"10376.0";"10317.0";"10880.0";"10457.0";
"12014.0";"9345.0";"10183
.0";"11072.0";"12263.0";
DEFUNCIONES;2003; Municipios de hasta 10.000
habitantes.Total;"107204.0";"10047.0";"8949.0";"8829.0";"8587.0";"8519.0";"8734.0";"887
9.0";"9684.0";"7585.0";"8275.0";"9090.0";"10026.0";
DEFUNCIONES;2003; Municipios de 10.001 a 50.000
habit.Total;"86743.0";"8140.0";"7312.0";"7432.0";"6857.0";"6762.0";"7122.0";"7075.0"
;"7578.0";"6239.0";"6634.0";"7301.0";"8291.0";
DEFUNCIONES;2003; Municipios > 50.000
habitantes.Total;"60045.0";"5456.0";"4948.0";"5066.0";"4776.0";"4734.0";"5080.0";"4877.0";
"545
3.0";"4400.0";"4767.0";"4907.0";"5581.0";
```



```
DEFUNCIONES;2003;  
DEFUNCIONES;2003;Copyright INE 2014  
DEFUNCIONES;2003;  
DEFUNCIONES;2003;  
DEFUNCIONES;2003;Paseo de la Castellana 183 - 28071 - Madrid - España Teléfono: (+34) 91  
583 91 00 - Contacta:www.ine.es/infoine
```

**Tabla 19: Formato de ficheros de nacimientos y defunciones pre procesados.**

Como se puede apreciar, se ha introducido al inicio de cada línea el tipo de datos que se van a cargar ya sean nacimientos o defunciones y el año de los datos. Esto es así para poder introducir como fichero de entrada un fichero genérico que procese a la vez los datos de nacimientos y defunciones y diferenciar a que tabla pertenecen desde dentro del mapeo. A su vez el año es necesario para introducirlo como parte de la clave del registro a insertar.

Otros ficheros procesados simultáneamente son los concernientes a los nacimientos y defunciones de padres extranjeros, estos ficheros previos a ser procesados tienen una cabecera y un pie de fichero. En el fichero se muestra la información de los nacimientos de padres extranjeros atendiendo a la provincia de residencia de los padres o el difunto, el mes de nacimiento o defunción y el sexo del recién nacido o el difunto. El fichero completo tiene el siguiente formato:

#### Nacimientos de padre o madre extranjera

Nacimientos. Año 2003

Nacimientos de madre extranjera por provincia de residencia de la madre , mes del nacimiento y sexo del nacido.

Units:Nacimientos

;Total;;;Enero;;;Febrero;;;Marzo;;;Abril;;;Mayo;;;Junio;;;Julio;;;Agosto;;;Septiembre;;;Octubre;;;  
Noviembre;;;Diciembre;;;

;Total;niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;  
niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;niños;niñas;Total;niños;  
niñas;Total;niños;niñas;Total;niños;niñas;

Total;"54028.0";"28014.0";"26014.0";"4170.0";"2120.0";"2050.0";"3804.0";"1953.0";"1851.0";  
"4142.0";"2161.0";"1981.0";"4034.0";"2100.0";"1934.0";"4372.0";"2269.0";"2103.0";"4460.0";  
"2316.0";"2144.0";"4785.0";"2472.0";"2313.0";"4735.0";"2460.0";"2275.0";"5012.0";"2589.0";  
"2423.0";"4880.0";"2509.0";"2371.0";"4765.0";"2489.0";"2276.0";"4869.0";"2576.0";"2293.0";  
Álava;"209.0";"105.0";"104.0";"14.0";"3.0";"11.0";"13.0";"6.0";"7.0";"15.0";"6.0";"9.0";"12.0";  
"9.0";"3.0";"25.0";"14.0";"11.0";"17.0";"7.0";"10.0";"24.0";"11.0";"13.0";"9.0";"6.0";"3.0";"21.  
0";"14.0";"7.0";"19.0";"11.0";"8.0";"19.0";"8.0";"11.0";"21.0";"10.0";"11.0";

Albacete;"311.0";"161.0";"150.0";"19.0";"11.0";"8.0";"27.0";"21.0";"6.0";"27.0";"14.0";"13.0";  
"20.0";"10.0";"10.0";"19.0";"13.0";"6.0";"24.0";"11.0";"13.0";"32.0";"13.0";"19.0";"26.0";"12.0  
";"14.0";"17.0";"6.0";"11.0";"34.0";"20.0";"14.0";"36.0";"17.0";"19.0";"30.0";"13.0";"17.0";

Alicante/Alacant;"2889.0";"1486.0";"1403.0";"223.0";"131.0";"92.0";"196.0";"99.0";"97.0";"20  
5.0";"101.0";"104.0";"231.0";"116.0";"115.0";"209.0";"100.0";"109.0";"227.0";"120.0";"107.0"  
;"281.0";"137.0";"144.0";"237.0";"119.0";"118.0";"250.0";"126.0";"124.0";"292.0";"159.0";"13  
3.0";"239.0";"116.0";"123.0";"299.0";"162.0";"137.0";

Almería;"1308.0";"693.0";"615.0";"109.0";"58.0";"51.0";"92.0";"50.0";"42.0";"106.0";"61.0";"4  
5.0";"103.0";"51.0";"52.0";"96.0";"44.0";"52.0";"123.0";"66.0";"57.0";"95.0";"44.0";"51.0";"13  
4.0";"79.0";"55.0";"113.0";"62.0";"51.0";"117.0";"59.0";"58.0";"105.0";"57.0";"48.0";"115.0";  
"62.0";"53.0";

Ávila;"74.0";"35.0";"39.0";"5.0";"2.0";"3.0";"9.0";"4.0";"5.0";"4.0";"1.0";"3.0";"6.0";"4.0";"2.0"  
;"6.0";"2.0";"4.0";"4.0";"3.0";"1.0";"8.0";"2.0";"6.0";"6.0";"3.0";"3.0";"6.0";"3.0";"3.0";"5.0";"3  
.0";"2.0";"5.0";"4.0";"1.0";"10.0";"4.0";"6.0";

Badajoz;"180.0";"106.0";"74.0";"21.0";"5.0";"16.0";"16.0";"9.0";"7.0";"10.0";"8.0";"2.0";"15.0"  
;"8.0";"7.0";"17.0";"11.0";"6.0";"12.0";"8.0";"4.0";"24.0";"15.0";"9.0";"17.0";"12.0";"5.0";"10.  
0";"7.0";"3.0";"11.0";"9.0";"2.0";"11.0";"6.0";"5.0";"16.0";"8.0";"8.0";

Balears

(Illes);"2130.0";"1076.0";"1054.0";"174.0";"92.0";"82.0";"153.0";"77.0";"76.0";"159.0";"80.0";  
"79.0";"158.0";"80.0";"78.0";"154.0";"66.0";"88.0";"174.0";"90.0";"84.0";"186.0";"95.0";"91.0  
";"202.0";"102.0";"100.0";"211.0";"106.0";"105.0";"181.0";"100.0";"81.0";"188.0";"91.0";"97.  
0";"190.0";"97.0";"93.0";

Barcelona;"8526.0";"4397.0";"4129.0";"593.0";"290.0";"303.0";"634.0";"329.0";"305.0";"671.  
0";"345.0";"326.0";"652.0";"327.0";"325.0";"714.0";"353.0";"361.0";"734.0";"379.0";"355.0";

726.0";"385.0";"341.0";"752.0";"376.0";"376.0";"743.0";"404.0";"339.0";"772.0";"398.0";"374.0";"772.0";"398.0";"374.0";"763.0";"413.0";"350.0";  
[...]

Source: Instituto Nacional de Estadística

Copyright INE 2014

Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 -  
Contacta:www.ine.es/infoine

**Tabla 20: Formato del fichero de nacimientos y defunciones de extranjeros.**

En el proceso de pre procesado se inserta al inicio de cada línea si ese registro es de un nacimiento o una defunción y el año de los datos que estamos procesando para que se pueda cruzar con las tablas maestras. Mediante el script de pre procesado insertamos los campos mencionados y realizamos una limpieza de los datos, cambiando el formato de los nombres de las provincias y dejándolos todos en el mismo formato ya que estos cambian de un año para otro. Una vez pre procesado el fichero, el formato que se obtiene es el siguiente:

```
DEFUNCIONES;2003;Defunciones_de_extranjeros
DEFUNCIONES;2003;__Defunciones._Año_2003
DEFUNCIONES;2003;
DEFUNCIONES;2003;Defunciones_de_extranjeros_por_provincia_de_residencia__mes_de_la_
defunción_y_sexo.
DEFUNCIONES;2003;Units:Defunciones
DEFUNCIONES;2003;
DEFUNCIONES;2003;
DEFUNCIONES;2003;;;Total;;;Enero;;;Febrero;;;Marzo;;;Abril;;;Mayo;;;Junio;;;Julio;;;Agosto;;;Se
ptiembre;;;Octubre;;;Noviembre;;;Diciembre;;;
DEFUNCIONES;2003;;;Total;Varones;Mujeres;Total;Varones;Mujeres;Total;Varones;Mujeres;To
tal;Varones;Mujeres;Total;Varones;Mujeres;Total;Varones;Mujeres;Total;Varones;Mujeres;Tot
al;Varones;Mujeres;Total;Varones;Mujeres;Total;Varones;Mujeres;Total;Varones;Mujeres;Tot
al;Varones;Mujeres;Total;Varones;Mujeres;
DEFUNCIONES;2003;Total;"8996.0";"5831.0";"3165.0";"789.0";"503.0";"286.0";"652.0";"418.0
";"234.0";"728.0";"480.0";"248.0";"712.0";"474.0";"238.0";"753.0";"493.0";"260.0";"737.0";"4
74.0";"263.0";"745.0";"498.0";"247.0";"745.0";"475.0";"270.0";"757.0";"514.0";"243.0";"764.0
";"492.0";"272.0";"770.0";"486.0";"284.0";"844.0";"524.0";"320.0";
DEFUNCIONES;2003;Álava;"15.0";"9.0";"6.0";"1.0";"1.0";"0";"1.0";"0";"1.0";"1.0";"1.0";"0";"2.
0";"1.0";"1.0";"3.0";"2.0";"1.0";"1.0";"0";"1.0";"0";"0";"0";"0";"0";"0";"0";"0";"0";"3.0";"2.0";"
1.0";"2.0";"1.0";"1.0";"1.0";"1.0";"0";
DEFUNCIONES;2003;Albacete;"9.0";"9.0";"0";"0";"0";"0";"0";"0";"2.0";"2.0";"0";"0";"0";"0
";"0";"0";"0";"0";"0";"0";"2.0";"2.0";"0";"0";"0";"0";"1.0";"1.0";"0";"1.0";"1.0";"0";"3.0";"3.0";
"0";"0";"0";"0";
DEFUNCIONES;2003;Alicante/Alacant;"1394.0";"899.0";"495.0";"115.0";"68.0";"47.0";"107.0";
"71.0";"36.0";"117.0";"71.0";"46.0";"112.0";"77.0";"35.0";"96.0";"60.0";"36.0";"100.0";"61.0";
"39.0";"125.0";"89.0";"36.0";"122.0";"88.0";"34.0";"120.0";"79.0";"41.0";"116.0";"75.0";"41.0
";"123.0";"71.0";"52.0";"141.0";"89.0";"52.0";
DEFUNCIONES;2003;Almería;"221.0";"150.0";"71.0";"21.0";"14.0";"7.0";"16.0";"10.0";"6.0";"1
4.0";"9.0";"5.0";"13.0";"11.0";"2.0";"15.0";"10.0";"5.0";"15.0";"10.0";"5.0";"26.0";"17.0";"9.0"
;"25.0";"16.0";"9.0";"15.0";"8.0";"7.0";"14.0";"11.0";"3.0";"18.0";"9.0";"9.0";"29.0";"25.0";"4.
0";
```

```

DEFUNCIONES;2003;Ávila;"8.0";"5.0";"3.0";"0";"0";"0";"0";"0";"1.0";"1.0";"0";"1.0";"0";"1.0";"0";"0";"0";"3.0";"1.0";"2.0";"0";"0";"0";"1.0";"1.0";"0";"1.0";"1.0";"0";"0";"0";"0";"0";"0";"1.0";"1.0";"0";
DEFUNCIONES;2003;Badajoz;"17.0";"11.0";"6.0";"1.0";"0";"1.0";"2.0";"0";"2.0";"2.0";"2.0";"0";"0";"0";"0";"1.0";"1.0";"0";"5.0";"3.0";"2.0";"2.0";"0";"1.0";"0";"1.0";"2.0";"2.0";"0";"1.0";"1.0";"0";"0";"0";"0";"0";"0";"0";"0";
DEFUNCIONES;2003;Balears_(Illes);"292.0";"187.0";"105.0";"23.0";"15.0";"8.0";"19.0";"14.0";"5.0";"23.0";"16.0";"7.0";"18.0";"13.0";"5.0";"25.0";"15.0";"10.0";"20.0";"13.0";"7.0";"30.0";"20.0";"10.0";"20.0";"10.0";"10.0";"33.0";"23.0";"10.0";"29.0";"13.0";"16.0";"30.0";"20.0";"10.0";"22.0";"15.0";"7.0";
DEFUNCIONES;2003;Barcelona;"487.0";"292.0";"195.0";"42.0";"26.0";"16.0";"41.0";"21.0";"20.0";"34.0";"18.0";"16.0";"39.0";"24.0";"15.0";"40.0";"18.0";"22.0";"36.0";"24.0";"12.0";"34.0";"21.0";"13.0";"41.0";"24.0";"17.0";"40.0";"27.0";"13.0";"34.0";"24.0";"10.0";"52.0";"31.0";"21.0";"54.0";"34.0";"20.0";
[...]
DEFUNCIONES;2003;
DEFUNCIONES;2003;Source:Instituto_Nacional_de_Estadística
DEFUNCIONES;2003;
DEFUNCIONES;2003;Copyright_INE_2014
DEFUNCIONES;2003;
DEFUNCIONES;2003;

```

Tabla 21: Formato del fichero de nacimientos y defunciones extranjeros pre procesado.

Los ficheros que contienen los datos de los censos de población informan al usuario de los habitantes resultantes en un determinado año atendiendo a su provincia y al tamaño de los municipios donde reside. El formato es el siguiente:

#### Resumen por Provincias

Cifras oficiales de población resultantes de la revisión del Padrón municipal a 1 de enero de 2003

Población por provincias y tamaño de los municipios.

Unidades: Personas

;Total; Menos de 101;de 101 a 500;De 501 a 1.000;de 1.001 a 2.000;De 2.001 a 5.000;De 5.001 a 10.000;de 10.001 a 20.000;De 20.001 a 50.000;De 50.001 a 100.000;de 100.001 a 500.000;Más de 500.000;

00

Total;"4.2717064E7";"57435.0";"717089.0";"774792.0";"1407178.0";"3090817.0";"3639128.0";"4781079.0";"6021723.0";"4927484.0";"9961028.0";"7339311.0";

01

Álava;"294360.0";"0.0";"5885.0";"10079.0";"13765.0";"12948.0";"9555.0";"18871.0";"0.0";"0.0";"223257.0";"0.0";

02

Albacete;"376556.0";"71.0";"5501.0";"15557.0";"34633.0";"50389.0";"23514.0";"14331.0";"77418.0";"0.0";"155142.0";"0.0";

03

Alicante/Alacant;"1632349.0";"201.0";"9431.0";"13629.0";"23960.0";"41277.0";"140029.0";"225766.0";"340919.0";"324063.0";"513074.0";"0.0";

04

Almería;"565310.0";"68.0";"9723.0";"10809.0";"21191.0";"51535.0";"71083.0";"66580.0";"42

```

514.0";"115080.0";"176727.0";"0.0";
05
Ávila;"165480.0";"3321.0";"33480.0";"18137.0";"8936.0";"30307.0";"19221.0";"0.0";"0.0";"52
078.0";"0.0";"0.0";
06
Badajoz;"663142.0";"76.0";"5133.0";"24496.0";"62062.0";"122085.0";"118959.0";"54531.0";"
85275.0";"52110.0";"138415.0";"0.0";
07 Balears
(Illes);"947361.0";"0.0";"695.0";"5714.0";"12677.0";"55379.0";"105843.0";"125973.0";"27380
3.0";"0.0";"367277.0";"0.0";
08
Barcelona;"5052666.0";"657.0";"16095.0";"23170.0";"53820.0";"160137.0";"331536.0";"4458
79.0";"621193.0";"752809.0";"1064632.0";"1582738.0";
09
[...]
```

Fuente: Instituto Nacional de Estadística

Copyright INE 2014

Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 -  
Contacta: [www.ine.es/infoine](http://www.ine.es/infoine)

Tabla 22: Formato del fichero de censos.

Una vez que se pre procesa el fichero se limpian los registros de espacios en blanco y se sustituyen por guiones bajos además, se modifican los nombres de las provincias que hayan podido sufrir alguna variación en el formato de un año para otro con el fin de que posteriormente crucen con la tabla maestra de provincias. Igualmente se añade al inicio de línea el año al que corresponden los datos. El fichero procesado quedaría de la siguiente manera:

```

2003;Resumen_por_Provincias
2003;__Cifras_oficiales_de_población_resultantes_de_la_revisión_del_Padrón_municipal_a_1
_de_enero_de_2003
2003;
2003;Población_por_provincias_y_tamaño_de_los_municipios.
2003;Unidades:Personas
2003;
2003;
2003;;Total;Menos_de_101;de_101_a_500;De_501_a_1.000;de_1.001_a_2.000;De_2.001_a_5
.000;De_5.001_a_10.000;de_10.001_a_20.000;De_20.001_a_50.000;De_50.001_a_100.000;d
e_100.001_a_500.000;Más_de_500.000;
2003;00_Total;"4.2717064E7";"57435.0";"717089.0";"774792.0";"1407178.0";"3090817.0";"3
639128.0";"4781079.0";"6021723.0";"4927484.0";"9961028.0";"7339311.0";
2003;01_Álava;"294360.0";"0.0";"5885.0";"10079.0";"13765.0";"12948.0";"9555.0";"18871.0"
;"0.0";"0.0";"223257.0";"0.0";
2003;02_Albacete;"376556.0";"71.0";"5501.0";"15557.0";"34633.0";"50389.0";"23514.0";"143
31.0";"77418.0";"0.0";"155142.0";"0.0";
2003;03_Alicante/Alacant;"1632349.0";"201.0";"9431.0";"13629.0";"23960.0";"41277.0";"140
029.0";"225766.0";"340919.0";"324063.0";"513074.0";"0.0";
2003;04_Almería;"565310.0";"68.0";"9723.0";"10809.0";"21191.0";"51535.0";"71083.0";"665
```



```

80.0";"42514.0";"115080.0";"176727.0";"0.0";
2003;05_Ávila;"165480.0";"3321.0";"33480.0";"18137.0";"8936.0";"30307.0";"19221.0";"0.0";
"0.0";"52078.0";"0.0";"0.0";
2003;06_Badajoz;"663142.0";"76.0";"5133.0";"24496.0";"62062.0";"122085.0";"118959.0";"5
4531.0";"85275.0";"52110.0";"138415.0";"0.0";
2003;07_Balears_(Illes);"947361.0";"0.0";"695.0";"5714.0";"12677.0";"55379.0";"105843.0";"
125973.0";"273803.0";"0.0";"367277.0";"0.0";
2003;08_Barcelona;"5052666.0";"657.0";"16095.0";"23170.0";"53820.0";"160137.0";"331536.
0";"445879.0";"621193.0";"752809.0";"1064632.0";"1582738.0";

```

[...]

2003;

2003;Fuente:Instituto\_Nacional\_de\_Estadística

2003;

2003;Copyright\_INE\_2014

2003;

2003;

2003;Paseo\_de\_la\_Castellana\_183\_-\_28071\_-\_Madrid\_-\_

\_España\_Teléfono:\_(+34)\_91\_583\_91\_00\_-\_Contacta:www.ine.es/infoine

**Tabla 23: Formato del fichero de censos pre procesado.**

Otro de los ficheros de entrada que procesaremos será el de población activa. Para éste caso realizaremos dos pre procesados en paralelo con el fin de crear un ejemplo más didáctico y que nos permita ver más funcionalidades posteriores de la herramienta y su manera de trabajo. En este caso procesaremos por un lado un fichero para un único año (2004) que tendrá un formato distinto y otro fichero procesado para el resto de años.

El fichero de datos para el año 2004 contiene los datos de población activa para ambos sexos y además el desglose por sexo. Estos datos están agregados por año, provincia y el estado o actividad económica. En este caso dado que los ficheros de los otros años no hacen distinción por sexo descartaremos desde el pre procesado los datos desglosados para hombres y mujeres de este fichero. Este fichero tendría el siguiente formato:

Resultados provinciales

Encuesta de población activa. Tablas anuales. Año 2004

Población de 16 y más años por sexo, provincia y relación con la actividad económica.

Unidades: miles de personas

;Población de 16 y más años;Activos;Ocupados;Totalparados;Parados que buscan primer empleo;Inactivos;

Ambos sexos;"";"";"";"";"";"";"";

Total;"35810.8";"20184.5";"17970.8";"2213.6";"336.2";"15626.3";

Álava;"252.8";"155.9";"141.4";"14.4";"1.6";"96.9";

Albacete;"309.5";"168.0";"152.3";"15.7";"1.7";"141.5";

Alicante;"1348.6";"783.6";"700.9";"82.7";"6.6";"565.0";

Almería;"472.7";"291.3";"261.8";"29.5";"3.0";"181.4";

Asturias;"937.9";"435.6";"390.4";"45.1";"11.6";"502.3";

Ávila;"140.6";"68.3";"61.8";"6.6";"1.0";"72.2";

Badajoz;"540.9";"280.6";"229.7";"50.9";"7.4";"260.3";

Balears (Illes);	"782.3";	"493.7";	"448.8";	"45.0";	"3.4";	"288.6";
Barcelona;	"4238.0";	"2557.9";	"2292.8";	"265.1";	"43.2";	"1680.1";
[...]						
Varones;	"";	"";	"";	"";	"";	"";
Total;	"17488.1";	"11905.1";	"10934.3";	"970.8";	"128.3";	"5583.1";
Álava;	"128.4";	"89.4";	"84.0";	"5.4";	"0.8";	"39.1";
Albacete;	"154.1";	"104.1";	"98.3";	"5.8";	"0.3";	"50.0";
Alicante;	"664.2";	"459.9";	"421.8";	"38.1";	"2.2";	"204.3";
Almería;	"230.3";	"172.8";	"162.2";	"10.6";	"1.5";	"57.5";
Asturias;	"446.0";	"256.7";	"238.3";	"18.4";	"4.4";	"189.2";
Ávila;	"73.0";	"44.9";	"42.0";	"2.9";	"0.3";	"28.1";
Badajoz;	"269.4";	"177.8";	"157.2";	"20.6";	"2.1";	"91.6";
Balears (Illes);	"388.7";	"283.2";	"261.8";	"21.3";	"1.5";	"105.5";
Barcelona;	"2048.0";	"1455.9";	"1333.8";	"122.0";	"22.9";	"592.1";
[...]						
Mujeres;	"";	"";	"";	"";	"";	"";
Total;	"18322.7";	"8279.4";	"7036.5";	"1242.9";	"207.9";	"10043.3";
Álava;	"124.3";	"66.5";	"57.4";	"9.1";	"0.8";	"57.8";
Albacete;	"155.4";	"63.9";	"54.0";	"9.9";	"1.4";	"91.5";
Alicante;	"684.4";	"323.6";	"279.1";	"44.6";	"4.5";	"360.8";
Almería;	"242.4";	"118.5";	"99.7";	"18.8";	"1.5";	"123.8";
Asturias;	"491.9";	"178.8";	"152.1";	"26.7";	"7.2";	"313.1";
Ávila;	"67.6";	"23.4";	"19.7";	"3.7";	"0.7";	"44.1";
Badajoz;	"271.5";	"102.8";	"72.6";	"30.3";	"5.3";	"168.7";
Balears (Illes);	"393.6";	"210.5";	"186.9";	"23.6";	"1.9";	"183.1";
Barcelona;	"2190.0";	"1102.1";	"959.0";	"143.1";	"20.2";	"1088.0";
[...]						
Notas:						
1.- Las estimaciones de Ceuta y Melilla no se publican desagregadas porque el tamaño muestral en cada una de ellas no lo permite						
Fuente:INE Instituto Nacional de Estadística (España)						
Copyright INE 2014						
Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 - Contacta:www.ine.es/infoine.						

Tabla 24: Fichero de ocupación de la población año 2004.

Una vez formateado el fichero se han descartado las líneas de datos con cifras de hombres y mujeres y se ha añadido al inicio de cada línea el año de los datos para poder cruzarlos con la tabla maestra. El formato sería el siguiente:

```

2004;Resultados_provinciales
2004;Encuesta_de_población_activa._Tablas_anuales._Año_2004
2004;
2004;Población_de_16_y_más_años_porsexo_provincia_y_relación_con_la_actividad_econó
mica.
2004;Unidades:miles_de_personas
2004;
2004;
2004;;Población_de_16_y_más_años;Activos;Ocupados;Total_parados;Parados_que_buscan_
primer_empleo;Inactivos;
2004;Ambos_sexos;"";"";"";"";"";"";
2004;Total;"35810.8";"20184.5";"17970.8";"2213.6";"336.2";"15626.3";
2004;Álava;"252.8";"155.9";"141.4";"14.4";"1.6";"96.9";
2004;Albacete;"309.5";"168.0";"152.3";"15.7";"1.7";"141.5";
2004;Alicante/Alacant;"1348.6";"783.6";"700.9";"82.7";"6.6";"565.0";
2004;Almería;"472.7";"291.3";"261.8";"29.5";"3.0";"181.4";
2004;Asturias;"937.9";"435.6";"390.4";"45.1";"11.6";"502.3";
2004;Ávila;"140.6";"68.3";"61.8";"6.6";"1.0";"72.2";
2004;Badajoz;"540.9";"280.6";"229.7";"50.9";"7.4";"260.3";
2004;Balears_(Illes);"782.3";"493.7";"448.8";"45.0";"3.4";"288.6";
2004;Barcelona;"4238.0";"2557.9";"2292.8";"265.1";"43.2";"1680.1";
[.]

```

**Tabla 25: Fichero de ocupación de la población pre procesado.**

Los ficheros para los años desde el 2005 hasta el 2013 tienen una particularidad que no tenía el fichero del 2004, estos ficheros solo tienen los datos de los dos sexos mezclados y además los datos no vienen agrupados por año si no que los datos de un año vienen separados por trimestres. Por ello el fichero tiene el siguiente formato antes de ser procesado.

Unidades:Miles de personas

;Total;;;Activos;;;Ocupados;;;Parados;;;Parados que buscan primer empleo;;;Inactivos;;;  
;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;20  
05TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005T  
II;2005TI;  
Ambos sexos;"",""  
02  
Albacete;"316.3";"315.1";"313.6";"312.4";"179.2";"175.3";"172.6";"173.3";"159.7";"156.3";"1  
60.0";"154.3";"19.5";"19.0";"12.6";"19.0";"2.6";"3.1";"2.0";"3.1";"137.1";"139.8";"141.0";"139  
.2";  
03  
Alicante/Alacant;"1425.6";"1409.7";"1396.0";"1382.6";"812.4";"818.0";"812.1";"811.8";"730.7  
";"747.0";"735.5";"728.2";"81.7";"70.9";"76.6";"83.6";"6.1";"8.6";"5.9";"8.1";"613.2";"591.7";"  
583.9";"570.8";  
04  
Almería;"500.1";"492.7";"487.5";"483.3";"312.9";"333.3";"300.8";"304.8";"283.7";"304.2";"27  
1.5";"277.8";"29.3";"29.1";"29.3";"27.0";"4.7";"2.9";"4.0";"6.9";"187.2";"159.4";"186.8";"178.

5";  
01  
Araba/Álava;"256.4";"255.8";"255.2";"254.5";"152.6";"145.1";"150.0";"151.9";"142.8";"132.9";  
;"141.8";"139.6";"9.8";"12.2";"8.2";"12.3";"1.0";"1.9";"0.7";"1.8";"103.8";"110.7";"105.3";"102  
.6";  
33  
Asturias;"937.4";"937.6";"937.7";"937.8";"457.0";"451.8";"448.4";"448.4";"413.3";"409.7";"39  
9.9";"397.8";"43.7";"42.1";"48.5";"50.6";"5.5";"8.6";"9.7";"7.3";"480.5";"485.8";"489.2";"489.  
4";  
05  
Ávila;"141.9";"141.5";"141.2";"141.0";"70.8";"72.5";"68.9";"68.2";"64.5";"66.1";"63.1";"61.9";  
"6.3";"6.4";"5.8";"6.3";"0.6";"0.9";"0.9";"1.5";"71.1";"69.0";"72.3";"72.8";  
06  
Badajoz;"546.3";"544.9";"543.8";"543.0";"278.5";"290.2";"280.2";"282.2";"231.9";"240.5";"23  
4.3";"226.4";"46.6";"49.7";"45.9";"55.8";"5.7";"7.6";"6.4";"8.0";"267.7";"254.7";"263.6";"260.  
8";  
07 Balears,  
Illes;"813.2";"807.1";"801.3";"795.9";"494.1";"526.3";"516.8";"480.6";"457.1";"499.9";"485.6"  
;"429.6";"37.0";"26.4";"31.2";"51.0";"2.5";"5.9";"4.5";"4.9";"319.1";"280.8";"284.6";"315.4";  
08  
Barcelona;"4338.8";"4317.7";"4297.5";"4280.8";"2692.7";"2617.2";"2620.6";"2581.2";"2516.0  
";"2458.2";"2434.6";"2370.8";"176.7";"159.0";"186.0";"210.5";"21.6";"21.4";"20.1";"25.2";"16  
46.0";"1700.5";"1676.9";"1699.6";  
[..  
Notas:  
1.- Los resultados de Ceuta y Melilla deben tomarse con precaución porque pueden estar  
afectados por grandes errores de muestreo.

Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 -  
Contacta:www.ine.es/infoine

**Tabla 26: Ficheros de ocupación de la población para los años 2005 al 2013.**

Una vez procesados estos ficheros se les añade al inicio de línea el año de los datos y se concatena el nombre de la provincia y el código numero con el guion bajo. El fichero pre procesado queda de la siguiente manera.

Unidades:Miles de personas

;Total;;;Activos;;;Ocupados;;;Parados;;;Parados que buscan primer empleo;;;Inactivos;;;  
;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;  
05TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005TII;2005TI;2005TIV;2005TIII;2005T  
II;2005TI;  
Ambos sexos;"";  
02  
Albacete;"316.3";"315.1";"313.6";"312.4";"179.2";"175.3";"172.6";"173.3";"159.7";"156.3";"1  
60.0";"154.3";"19.5";"19.0";"12.6";"19.0";"2.6";"3.1";"2.0";"3.1";"137.1";"139.8";"141.0";"139  
.2";  
03  
Alicante/Alacant;"1425.6";"1409.7";"1396.0";"1382.6";"812.4";"818.0";"812.1";"811.8";"730.7

","747.0","735.5","728.2","81.7","70.9","76.6","83.6","6.1","8.6","5.9","8.1","613.2","591.7","  
 583.9","570.8";  
 04  
 Almería,"500.1","492.7","487.5","483.3","312.9","333.3","300.8","304.8","283.7","304.2","27  
 1.5","277.8","29.3","29.1","29.3","27.0","4.7","2.9","4.0","6.9","187.2","159.4","186.8","178.  
 5";  
 01  
 Araba/Álava,"256.4","255.8","255.2","254.5","152.6","145.1","150.0","151.9","142.8","132.9"  
 ;"141.8","139.6","9.8","12.2","8.2","12.3","1.0","1.9","0.7","1.8","103.8","110.7","105.3","102  
 .6";  
 33  
 Asturias,"937.4","937.6","937.7","937.8","457.0","451.8","448.4","448.4","413.3","409.7","39  
 9.9","397.8","43.7","42.1","48.5","50.6","5.5","8.6","9.7","7.3","480.5","485.8","489.2","489.  
 4";  
 05  
 Ávila,"141.9","141.5","141.2","141.0","70.8","72.5","68.9","68.2","64.5","66.1","63.1","61.9";  
 "6.3","6.4","5.8","6.3","0.6","0.9","0.9","1.5","71.1","69.0","72.3","72.8";  
 06  
 Badajoz,"546.3","544.9","543.8","543.0","278.5","290.2","280.2","282.2","231.9","240.5","23  
 4.3","226.4","46.6","49.7","45.9","55.8","5.7","7.6","6.4","8.0","267.7","254.7","263.6","260.  
 8";  
 07 Balears,  
 Illes,"813.2","807.1","801.3","795.9","494.1","526.3","516.8","480.6","457.1","499.9","485.6"  
 ;"429.6","37.0","26.4","31.2","51.0","2.5","5.9","4.5","4.9","319.1","280.8","284.6","315.4";  
 08  
 Barcelona,"4338.8","4317.7","4297.5","4280.8","2692.7","2617.2","2620.6","2581.2","2516.0  
 ","2458.2","2434.6","2370.8","176.7","159.0","186.0","210.5","21.6","21.4","20.1","25.2","16  
 46.0","1700.5","1676.9","1699.6";  
 [..]  
 2005;  
 2005;Notas:  
 2005;\_1.-  
 \_Los\_resultados\_de\_Ceuta\_y\_Melilla\_deben\_tomarse\_con\_precaución\_porque\_pueden\_esta  
 r\_afectados\_por\_grandes\_errores\_de\_muestreo.\_  
 2005;  
 2005;  
 2005;  
 2005;Paseo\_de\_la\_Castellana\_183\_-\_28071\_-\_Madrid\_-\_  
 \_España\_Teléfono:(+34)\_91\_583\_91\_00\_-\_Contacta:www.ine.es/infoine

Tabla 27: Ficheros de ocupación de la población para los años 2005 al 2013 pre procesados.

Estos datos desagrupados por trimestres se agruparan y sumaran posteriormente desde la herramienta Powercenter, durante el proceso de carga que más adelante explicaremos.

Por último tenemos otro proceso de carga que compara las rentas medias por sectores y comunidades entre los años 2004 y 2011. El fichero con los datos en bruto tiene la siguiente estructura.

Resultados por CCAA: Ganancia media anual por trabajador  
Encuesta de estructura salarial. Serie 2004-2007

Sexo y Sectores de actividad

Unidades:euros

;Ambos sexos;;;Mujeres;;;Varones;;;

:2004:2005:2006:2007;2004:2005:2006:2007;2004:2005:2006:2007:

Total Nacional: ""."".""."".""."".""."".""."".""."".""."

Todos los sectores de

actividad;"18310.11";"18676.92";"19680.88";"20390.35";"14889.74";"15294.83";"16245.17";"16943.89";"20548.33";"21093.92";"22051.08";"22780.29";

Industria;"21150.86";"21998.4";"22635.53";"23139.85";"16103.18";"16675.05";"17532.07";"18267.85";"22900.85";"23913.66";"24328.47";"24828.86";

Construcción;"17226.13";"17568.66";"17848.2";"18693.17";"15047.33";"15484.32";"15635.65";"16580.27";"17414.66";"17765.14";"18048.86";"18922.38";

Servicios;"17728.6";"18071.07";"19249.35";"19980.14";"14703.86";"15111.28";"16095.14";"16775.5";"20880.28";"21387.29";"22684.06";"23391.53";

Andalucía;,,,,,,;

Todos los sectores de

actividad;"16315.94";"16658.41";"17372.42";"17987.33";"13248.48";"13738.98";"14137.55";"14398.98";"17953.63";"18265.03";"19253.11";"20042.29";

Industria;"18323.2";"18837.81";"20053.22";"20588.35";"13755.22";"13304.47";"14493.79";"15021.54";"19634.01";"20511.34";"21447.74";"22032.27";

Construcción;"16918.43";"17355.63";"16923.79";"17634.69";"..";"-16167.97";"-13215.29";"..";"17149.22";"17453.12";"17221.37";"17964.74";

Servicios;"15709.27";"16006.78";"16979.42";"17579.64";"13205.44";"13647.41";"14148.2";"14370.04";"17894.87";"18125.04";"19600.32";"20428.0";

Aragón: "";"";"";"";"";"";"";"";"";"";"";"";

Todos los sectores de

actividad;"18046.13";"18592.08";"19134.82";"20015.81";"14020.33";"14325.81";"14894.41";"15694.24";"20613.66";"21479.67";"22030.14";"22771.23";

Industria;"20337.52";"20987.94";"22038.7";"22421.73";"15023.25";"14923.57";"16143.89";"16772.37";"22101.7";"22932.76";"23900.9";"24155.51";

Construcción;"17986.07";"18802.18";"19353.02";"20020.17";"..";"..";"..";"..";"-18370.23";"-19327.55";"19562.33";"20376.93";

Servicios;"16998.51";"17481.6";"17733.96";"18931.68";"13796.67";"14189.37";"14572.68";"15405.08";"20444.89";"21373.55";"21674.53";"22793.36";

Asturias (Principado de);"";"";"";"";"";"";"";"";"";"";"";"".

Todos los sectores de

actividad;"18395.99";"18882.69";"19758.43";"20623.81";"13652.65";"14034.25";"15356.82";

16376.95";"21312.4";"21904.34";"22781.54";"23347.87";  
 Industria;"24504.1";"24827.57";"25275.31";"25281.29";"-16192.71";"-  
 16983.67";"16925.02";"-17900.64";"25916.05";"26200.84";"26596.11";"26592.36";  
 Construcción;"20058.23";"20528.1";"20208.17";"20688.87";".."..".."20314.47";"2083  
 4.54";"20310.27";"20846.77";

Servicios;"16003.49";"16622.75";"17847.31";"19219.72";"13317.34";"13725.8";"15148.17";"1  
 6142.96";"19001.5";"19949.23";"21447.69";"22669.44";  
 Balears (Illes);""""""""""""""""""""

Todos los sectores de  
 actividad;"16635.58";"17018.24";"18095.93";"18886.8";"14290.28";"14819.07";"15750.0";"16  
 627.15";"18395.55";"18696.42";"20018.29";"20846.75";

Industria;"17286.57";"18101.47";"19289.99";"20042.82";"-13310.67";"-13828.42";"-  
 15036.93";"-15995.16";"18633.34";"19816.31";"20767.03";"21437.8";

Construcción;"-15667.58";"-16153.45";"16666.19";"-17129.86";".."..".."20314.47";  
 15821.05";"-16118.82";"16964.85";"-17456.24";

Servicios;"16753.08";"17077.82";"18252.62";"19089.97";"14339.82";"14817.66";"15849.17";"  
 16733.3";"19254.08";"19442.14";"21053.86";"21950.17";

Canarias;""""""""""""""""""""

Todos los sectores de  
 actividad;"15253.6";"15590.68";"16431.12";"17189.07";"13882.8";"13728.92";"14840.03";"15  
 721.06";"16142.03";"16771.82";"17535.2";"18196.14";

Industria;"18024.94";"18061.37";"18340.49";"18598.86";"-14261.98";"-14016.89";"-  
 14405.77";"-14919.46";"18970.5";"19110.9";"19315.45";"19384.93";

Construcción;"14794.94";"15393.14";"15016.87";"15632.52";".."..".."14866.25";"153  
 81.57";"15035.08";"15591.74";

Servicios;"15120.94";"15446.38";"16554.24";"17363.63";"13873.87";"13638.93";"14855.7";"1  
 5734.1";"16309.59";"17122.94";"18213.62";"18946.29";

[...]

Notas:

1.- Cuando la casilla está marcada con un signo '-' antes del dato, indica que el número de  
 observaciones muestrales está comprendido entre 100 y 500, por lo que la cifra es poco  
 significativa.

2.- '..'=No se facilita el dato correspondiente por ser el número de observaciones muestrales  
 inferior a 100

Fuente:INE: La información del año 2006 se ha extraído de la Encuesta de Estructura Salarial  
 2006.

Copyright INE 2014

Paseo de la Castellana, 183 - 28071 - Madrid - España Teléfono: (+34) 91 583 91 00 -  
 Contacta:www.ine.es/infoine

Tabla 28: Fichero de rentas por sector y comunidad.





### 3.4 Carga de datos a través de POWERCENTER.

En este punto procederemos a detallar cada uno de los procesos de carga de los cuales hemos preparado anteriormente los ficheros de entrada.

#### 3.4.1 Carga de datos para nacimientos y defunciones nacionales.

En este proceso de carga realizaremos la carga simultánea en sus respectivas tablas de destino de nacimientos y defunciones de los ficheros anteriormente pre procesados.

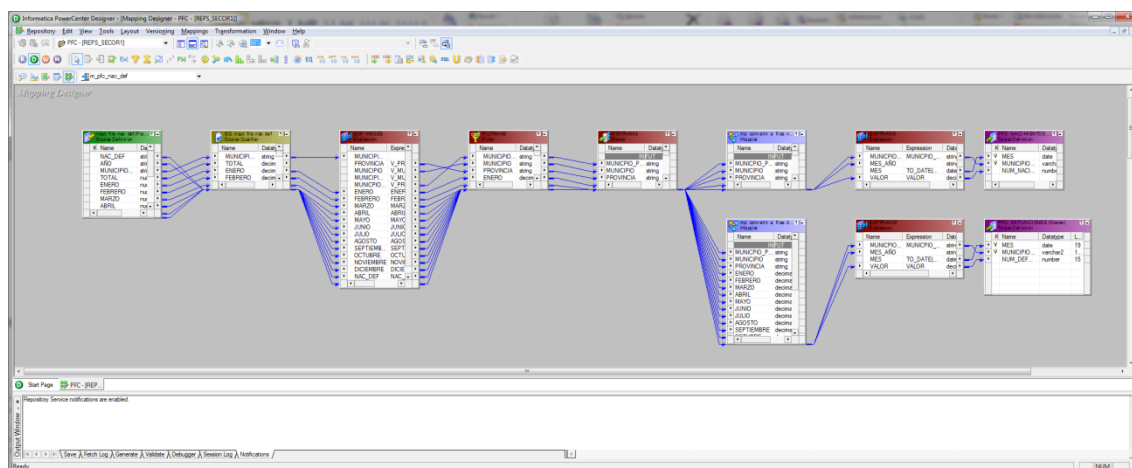


Ilustración 133: Proceso de carga visto globalmente.

Inicialmente realizamos una lectura de los ficheros de nacimientos y defunciones, ambos tienen la misma estructura. De esta manera nos hemos asegurado la posibilidad de procesar ambos ficheros simultáneamente sin hacer distinciones a la hora de su lectura.

Para la lectura de cualquier entrada ya sean ficheros o tablas de bbdd utilizamos dos componentes, el input del fichero y un componente llamado “sourcequalifier”. Este último, recoge la lectura del fichero o de la tabla que estemos leyendo, pudiendo realizar algunos filtros o modificaciones en tiempo de lectura sobre los datos que vamos a continuar procesando. En nuestro caso, es una lectura integra del fichero sin ningún tipo de filtro aplicado en el momento de su lectura sobre los datos.

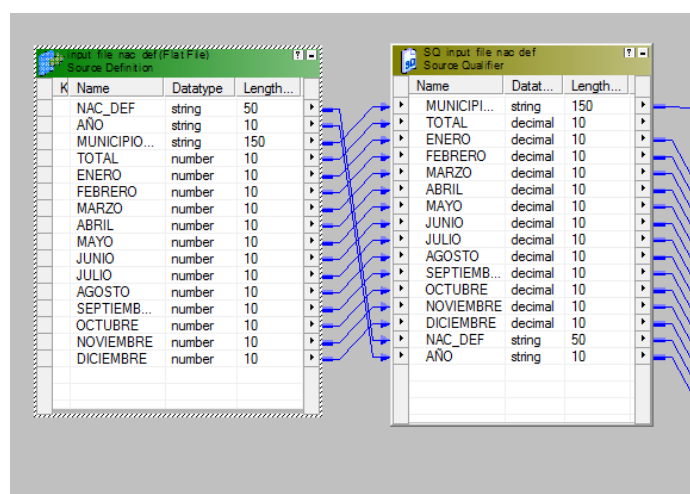


Ilustración 134: Lectura de ficheros.

Una vez hemos leído los registros de todos los ficheros, estos pasan a un componente denominado “expresion”, el cual permite realizar transformaciones y aplicarle distintas funciones sobre los datos. En nuestro caso, recibimos los datos de los registros leídos y pre procesamos el campo municipio\_provincia\_in de entrada realizando sobre él varias transformaciones para obtener finalmente el campo municipio y provincia tal cual tenemos en nuestra tabla maestra y no como aparece en el fichero.

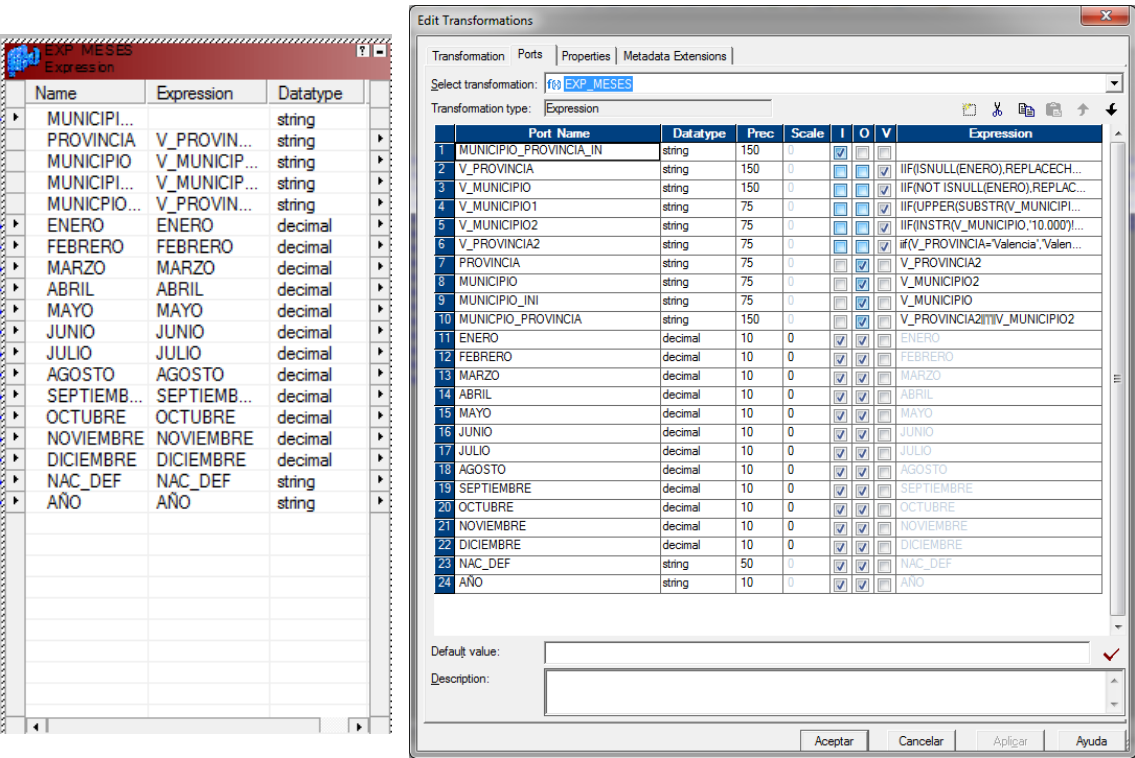


Ilustración 135: Apariencia del componente expresión y pantalla de configuración.

Para ello nos ayudamos de unos campos intermedios denominados variables y que se identifican marcando un tic en la columna “V” de la caja del componente, este campo variable solo tiene validez dentro del ámbito de la caja de la expresión. En un primer momento, comprobamos si del registro leído, el campo perteneciente al mes de enero es nulo de ser así implicaría que estamos ante un registro que solo contiene el nombre de la provincia por lo que cogeríamos ese campo municipio\_provincia y lo guardaríamos en la variable “v\_provincia”, de no ser nulo implicaría que estamos ante un registro de datos por municipio y si que contendría los datos de los meses de nacimientos o defunciones por lo que la mantendríamos el valor de la variable provincia que tuviéramos anteriormente.

El registro tal cual nos viene del fichero sería con este formato en caso de que el mes de enero fuera nulo y nos encontraríamos ante un registro de provincia:

NAC_DEF	ANYO	MUNICIPIO_PROVINCIA	TOTAL	ENERO	FEBRERO	MARZO	ABRIL
DEFUNCIONES	2003	Álava	NULL	NULL	NULL	NULL	NULL

MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabla 30: Registro de provincia.

En caso de que no fuera null nos encontraríamos ante un registro de datos y tendría el siguiente formato:

NAC_DEF	ANYO	MUNICIPIO_PROVINCIA	TOTAL	ENERO	FEBRERO	MARZO
DEFUNCIONES	2003	Municipios de hasta 10.000 habitantes.Total	517.0	46.0	34.0	55.0

ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
47.0	34.0	41.0	35.0	42.0	39.0	43.0	51.0	50.0

Tabla 31: Registro con datos mensuales.

Tras aplicar la fórmula que se muestra en la siguiente ilustración obtendríamos el nombre de la provincia limpio, sin espacios en blanco delante o detrás y habiendo sustituido los espacios en blanco intermedios por guiones bajos, de no ser null. La fórmula aplicada tendría el siguiente formato:

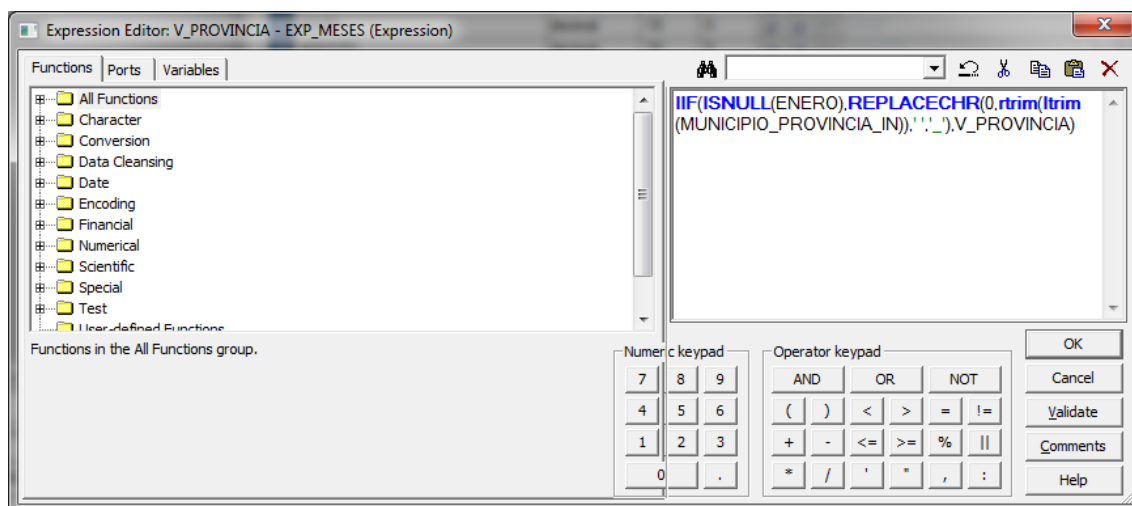


Ilustración 136: Obtención y limpieza de la variable provincia.

De igual manera realizamos una vez obtenida la provincia una nueva transformación apoyándonos en otra variable **V\_PROVINCIA2**, en la cual comprobamos si hemos obtenido en la variable **V\_PROVINCIA** el valor 'Valencia' y de ser correcto le cambiamos por 'Valencia/València', que es el valor que tenemos contenido en nuestra tabla maestra. En caso

de no coincidir este valor se mantiene el resultado que tuviéramos. La expresión realizada para esta transformación sería la siguiente:

```
iif(V_PROVINCIA='Valencia','Valencia/València',V_PROVINCIA)
```

Para la obtención del municipio realizaremos varias operaciones con el campo municipio provincia en aquellos registros que contengan datos, como el que se muestra en la tabla 27, para ello nos ayudaremos de dos variables intermedias.

**Nombre variable:** V\_MUNICIPIO.

**Función:**

```
IIF(NOT ISNULL(ENERO),REPLACECHR(0, rtrim(ltrim(MUNICIPIO_PROVINCIA_IN)),',','_'),null)
```

**Explicación:** Si el campo enero del registro leído no es nulo, cogemos el contenido del campo municipio\_provincia y se eliminan los posibles espacios en blanco del inicio y fin, además sustituimos los espacios en blanco intermedios por guiones bajos. En caso de que sea nulo el campo enero, significará que es un campo de provincia y esta variable permanecerá a null.

**Nombre variable:** V\_MUNICIPIO2.

**Función:**

```
IIF(INSTR(V_MUNICIPIO,'10.000')!=0,'menos_10000',  
iif(INSTR(V_MUNICIPIO,'10.001')!=0,'10001-50000',  
iif(INSTR(V_MUNICIPIO,'>_50.000')!=0 OR INSTR(V_MUNICIPIO,'de_50.000')!=0,'mas_50000',  
IIF(UPPER(V_MUNICIPIO)=UPPER('Capital'),'capital',''))))
```

**Explicación:** En este caso comprobamos de que tipo de municipios se trata según su número de habitantes, para ello comprobamos los textos literales que vienen en el fichero y si cumplen que contienen un valor de nuestro intervalo de habitantes de un municipio le renombramos con el valor que tenemos alojado en nuestra tabla maestra de municipios. Estos valores pueden ser: 'menos\_10000', '10001-50000', 'mas\_50000' y 'capital'.

Finalmente obtenemos el campo final MUNICIPIO\_PROVINCIA como la concatenación de las variables V\_PROVINCIA2 y V\_MUNICIPIO2, con la siguiente expresión  
V\_PROVINCIA2||'|'||V\_MUNICIPIO2.

Una vez obtenido el registro con el formato que queremos para cada campo realizaremos un filtrado de los registros para eliminar los que no nos interesan, en este caso dejaremos pasar los registros que tengan el campo no contengan el campo municipio a vacío y los registros cuyo campo municipio\_provincia no contenga la palabra extranjero.

**Expresión del filtro:**

```
MUNICIPIO != " and substr(MUNICIPIO_PROVINCIA,0,instr(MUNICIPIO_PROVINCIA,'|')-1) !=  
'Extranjero'
```

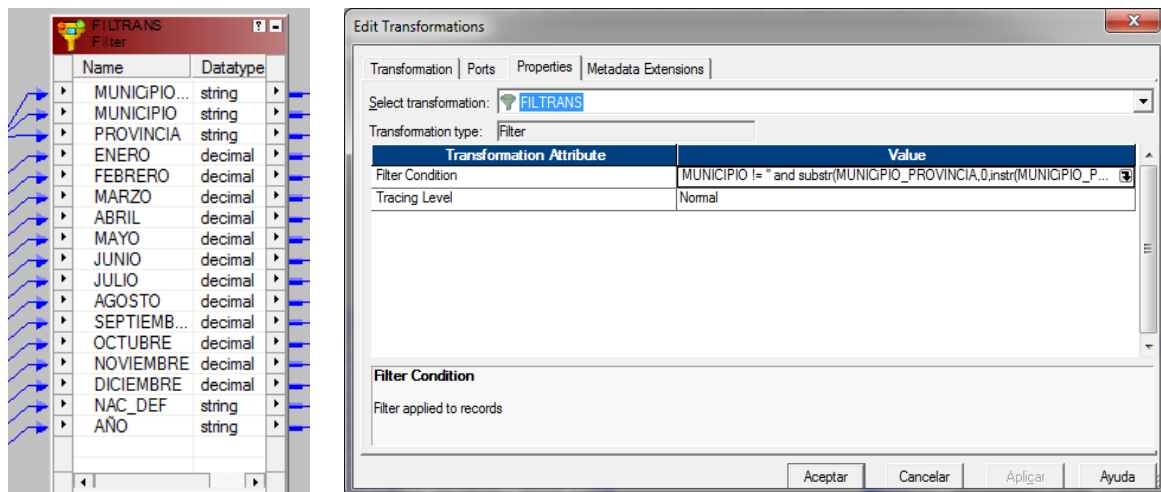


Ilustración 137: Componente filtro y configuración del filtro.

Una vez filtrados estos registros realizaremos una bifurcación para separar los registros que correspondan a nacimientos y los que correspondan a defunciones, para ello nos ayudaremos de un componente denominado “Router” que realiza la misma función que realizaría un if o un switch en java.

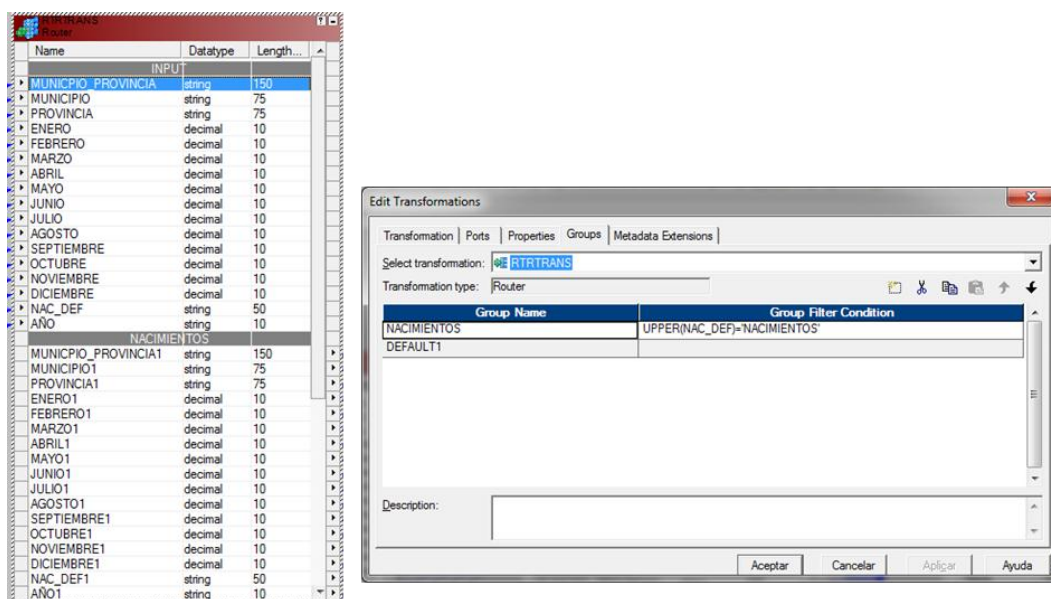


Ilustración 138: Componente router y su configuración

En este caso para dirigir los registros en un sentido u otro se comprueba el campo NAC\_DEF que se ha cargado durante el preprocesado con los valores “DEFUNCIONES” o “NACIMIENTOS”, si es nacimientos se dirigirá por la rama de nacimientos en caso contrario será una defunción e ira por la via por defecto ( aquí se podría haber creado otro grupo comprobando las defunciones, lo cual haria el proceso mas generico, pero como hemos preprocesado nosotros el fichero y nos hemos asegurado que o es nacimiento o es defunción y no hay otro posible valor he optado por evitar otra comprobación más por cada registro con el fin de hacer el proceso de carga mas ágil).

Una vez bifurcado el flujo de los registros cada registro se dirige a un mapplet que realizará la operación de trasponer un registro de varias columnas en varios registros con una misma clave y cada columna de meses del registro original formará un nuevo registro.

En el mapeo principal el flujo se representaría con la salida desde el router y las entradas a ambos mapplets de la siguiente forma.

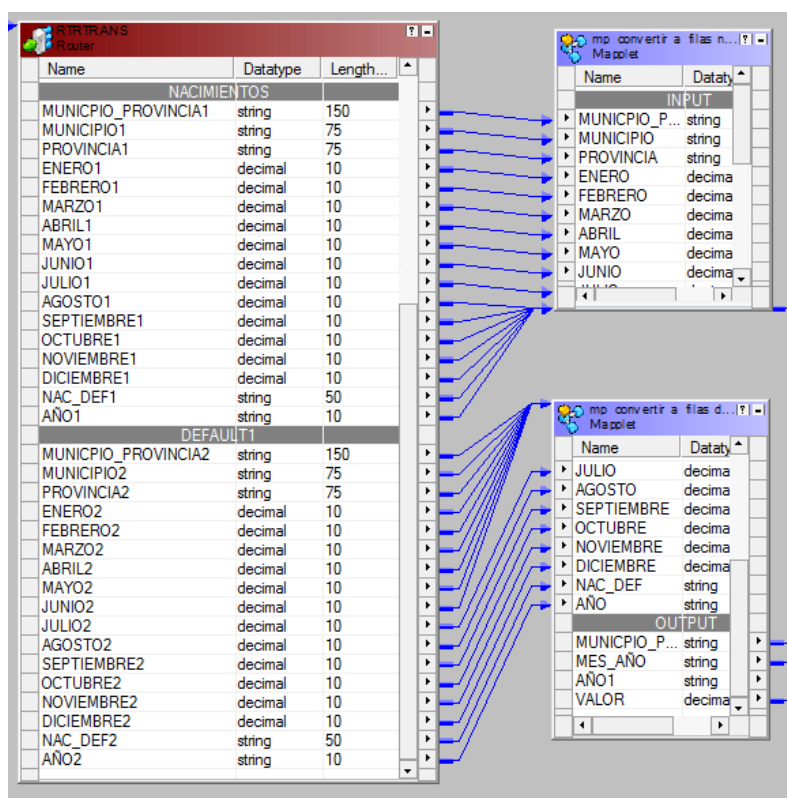


Ilustración 139: Entrada de los registros a los mapplets en el flujo principal de carga.

Dentro del mapplet tendremos el siguiente algoritmo de carga:

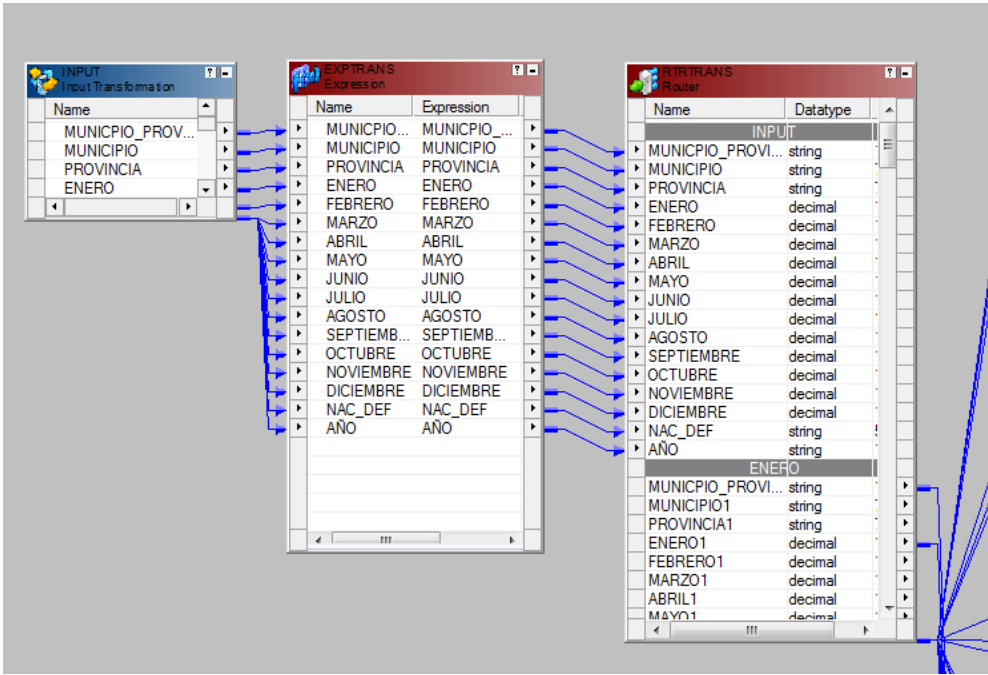


Ilustración 140: Entrada de datos en el mapplet.

En primer lugar se tiene el componente de entrada de registros que simplemente tiene la estructura que tendrán los registros que recibiremos en el mapplet.

Estos campos se reciben como entrada en una expresión que servirá de entrada a su vez para un router que tendrá un grupo por cada mes del año, en este caso no se aplica ninguna comprobación dentro de los grupos del router de tal manera que cada registro que llegue puede salir por cada grupo que decidamos con los datos que decidamos sacar, de esta manera forzamos la creación de un producto cartesiano de cada registro. Así por cada registro que llegue al router tendrá 12 salidas, una por cada mes del año pero esta vez en lugar de sacar todos los datos de todos los meses, saldrá la clave del registro (municipio\_provincia, año y mes) y el valor del mes concreto que estemos tratando en ese grupo.

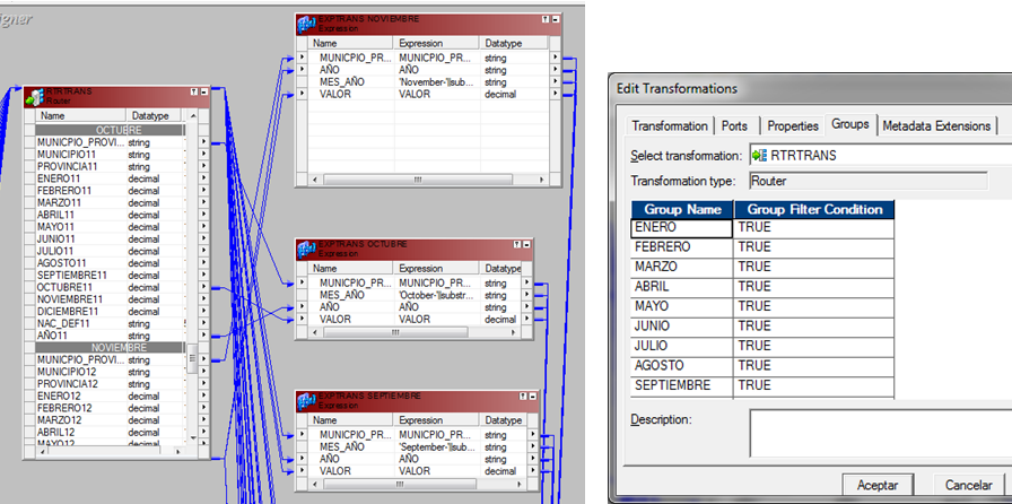
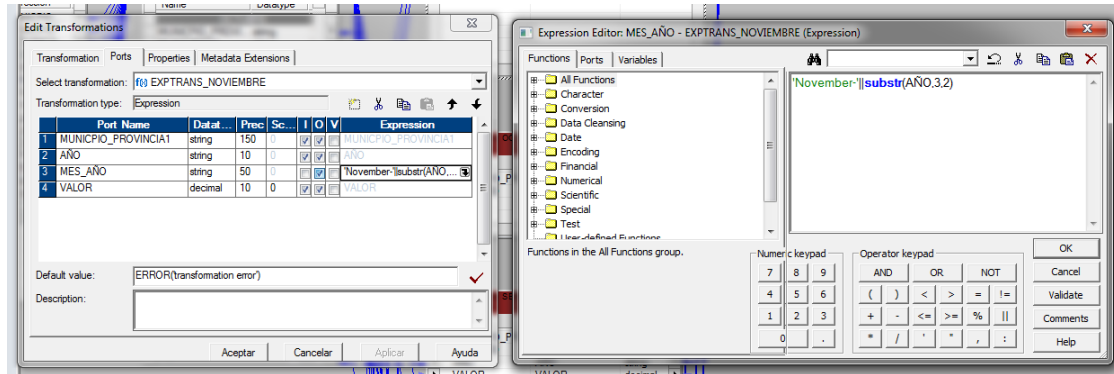


Ilustración 141: Bifurcación de datos en meses y configuración del router

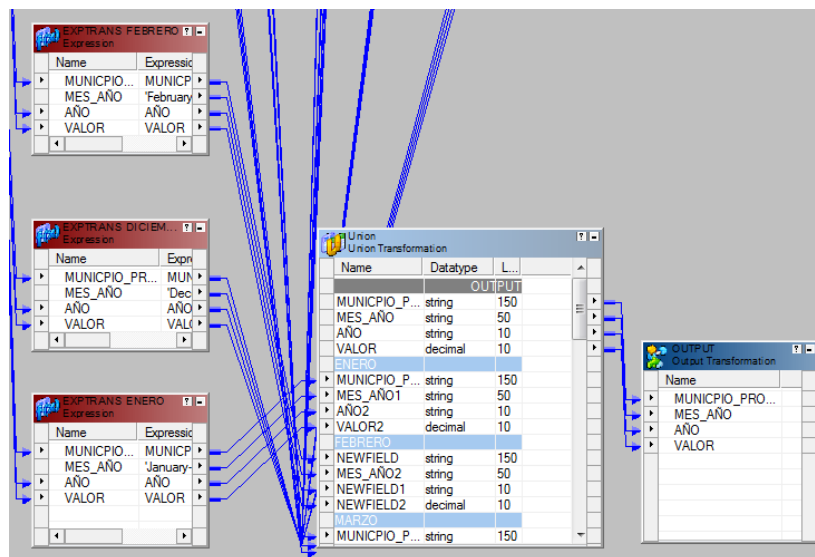


Una vez conseguido el cartesiano, los nuevos registros se pasan a una expresión por cada uno de los meses y en cada una de ellas se creará el campo mes\_año como la concatenación de el año escrito como literal en ingles (esto es así porque la bbdd que tenemos configurada, las fechas las tiene configuradas en ingles y no en castellano) mas los dos últimos dígitos del campo año que recibimos como entrada (función aplicada: 'November-'||**substr**(AÑO,3,2)).



**Ilustración 142: Configuración de las expresiones mensuales y funciones aplicadas en el cálculo del nuevo campo.**

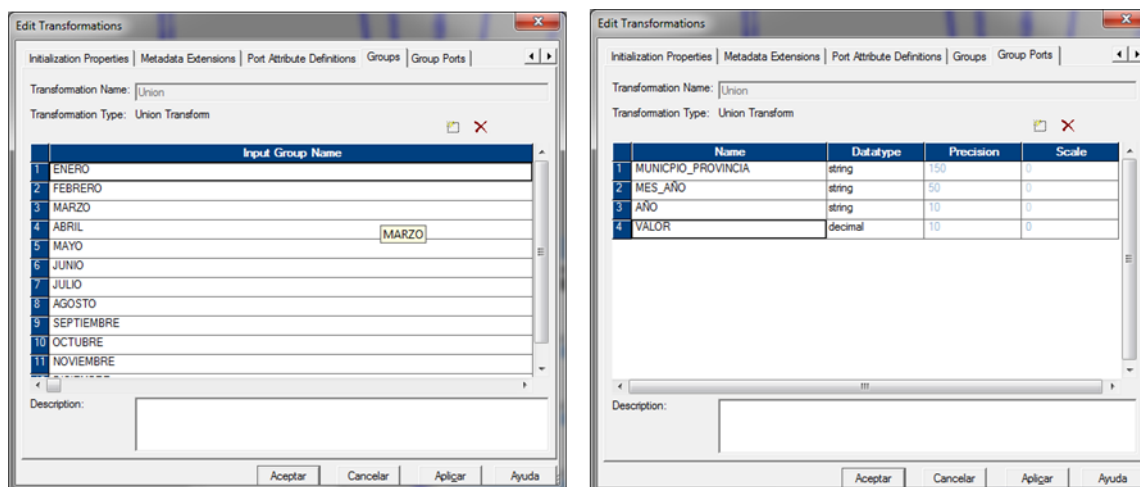
Una vez que ya hemos realizado el cartesiano y creado todos los campos mes\_año de los nuevos registros mediante un componente UNION volvemos a unificar el flujo que habíamos bifurcado mediante el router en un único flujo. Esta vez el componente UNION recibirá doce entradas distintas, una por cada mes del año y sacará una única salida.



**Ilustración 143: Unificación de flujos y salida del maplet.**



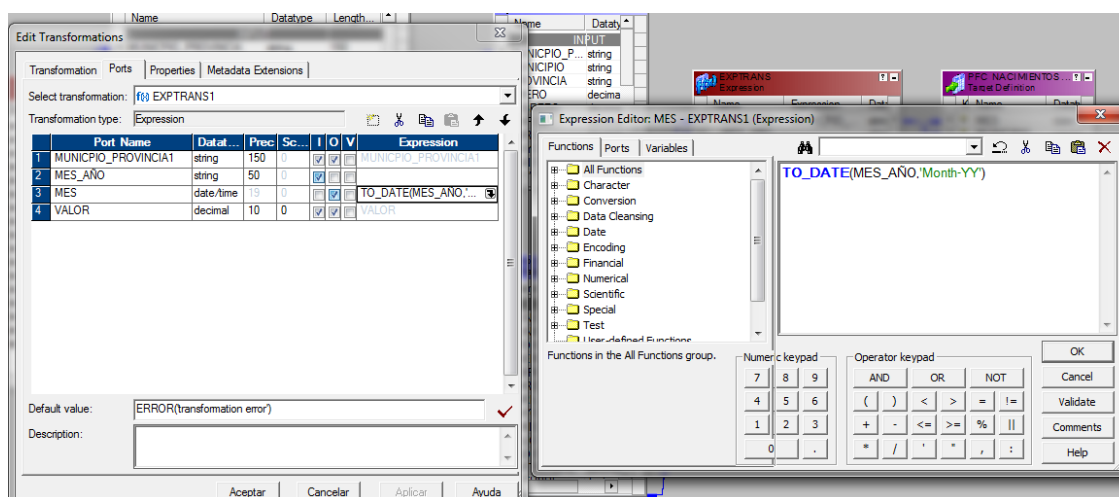
Para utilizar el componente UNION al igual que en sql en Oracle se tiene que cumplir que cada grupo de entrada del UNION tenga el mismo número de campos y la misma estructura. La configuración de los grupos y estructura de los campos en el componente UNION de Powercenter sería como se muestra en la siguiente imagen.



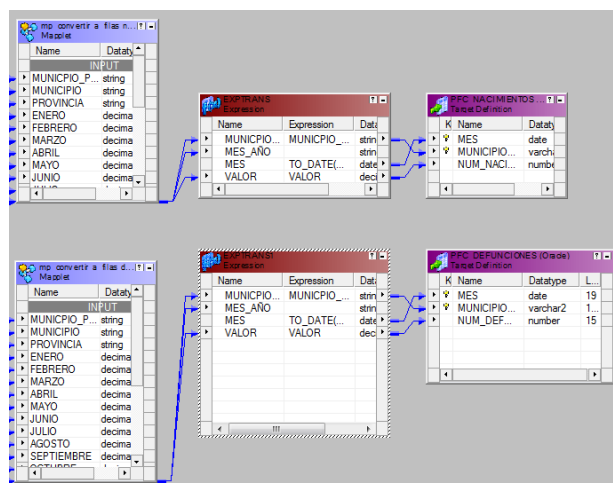
**Ilustración 144: Configuración del componente UNION.**

Una vez reunificado el flujo los nuevos registros saldrán a través del componente de salida del mapplet mostrado en la ilustración 144.

De vuelta al mapeo principal tendremos el flujo dividido en dos, un flujo para cada tabla de destino bien sean nacimientos o bien sean defunciones. La salida del mapplet es recibida en cada flujo por una expresión que transformará el campo mes\_año calculado en el mapplet y que era de tipo cadena de texto (string) en un campo de tipo fecha mediante la función to\_date y una vez creado este campo se cargan los registros en las tablas de bbdd.



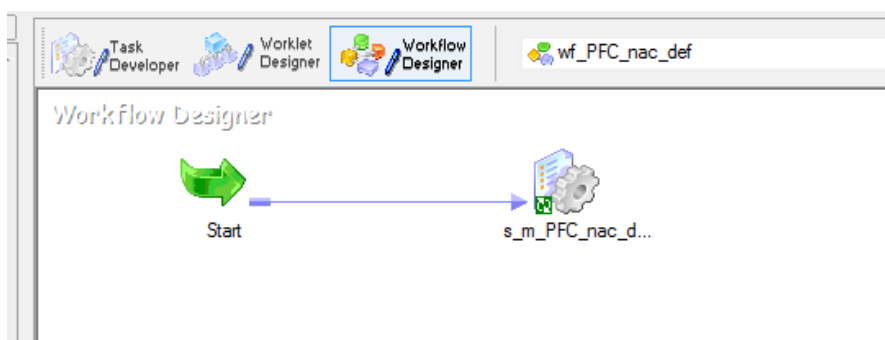
**Ilustración 145: Creación y configuración del nuevo campo fecha.**



**Ilustración 146: Finalización del proceso de carga.**

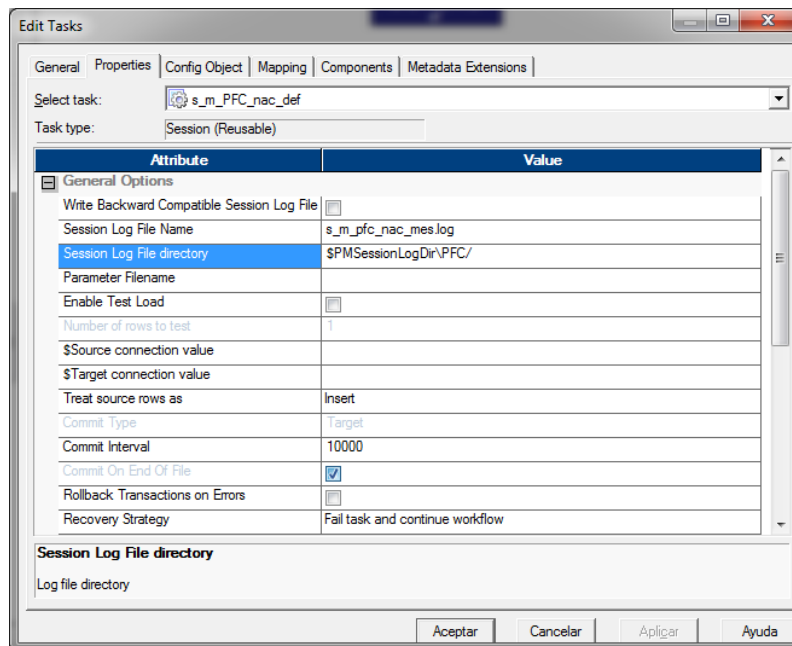
A mas alto nivel, una vez que “instanciamos” el mapping y creamos el workflow, en este caso el workflow se compone de una única sesión que instancia el mapping anteriormente descrito. La única configuración relevante de este workflow sería la definición de las rutas de logs y las definiciones de orígenes y destinos. A continuación explicaremos estos puntos.

La vista global del workflow seria la siguiente:



**Ilustración 147: Vista general del workflow.**

La configuración de los se realiza entrando en la sesión y situándonos en la pestaña de “Properties”. Aquí tendremos que rellenar los campos Session Log File Name con el nombre del fichero que queramos ponerle al log y por otro lado, tendremos que rellenar el campo Session Log File directory con el path de la ruta donde queremos dejar el fichero de logs.



**Ilustración 148: Configuración de los logs de sesión.**

La parte más importante de la configuración de las sesiones es la que se indica en la pestaña “Mapping”, aquí se configura a que ficheros o tablas y conexiones de bbdd apuntan cada uno de nuestros orígenes y destinos.

En el caso de los orígenes se tendrá que indicar que tipo de origen es, si es un fichero o una tabla de bbdd, en nuestro caso es un fichero lo que recibimos como entrada y por ello seleccionamos la opción “File Reader”. Las otras opciones que hay que configurar en las entradas de ficheros son el campo “Sourcefiletype” en nuestro caso es indirecto (esto significa que el fichero que le indiquemos que recibiremos como entrada, no contendrá los datos propiamente dichos que vayamos a cargar, si no que contendrá una lista de ficheros incluidos los paths donde están ubicados, y que la herramienta tendrá que procesar cada uno de esos ficheros que vienen indicados en el fichero de entrada. Si seleccionáramos la opción direct, significaría que el nombre del fichero que le indiquemos como entrada contendría directamente los datos que necesitamos cargar) y el campo “Sourcefilename” nosotros borramos el contenido del campo “Source file directory” puesto que el fichero que le introducimos como parámetro en el campo “Sourcefilename” no solo contiene el nombre del fichero si no que contiene la ruta absoluta hasta este fichero.

Como se puede ver no hemos puesto la ruta absoluta y el nombre del fichero si no que hemos puesto el nombre de un parámetro que se aloja en un fichero que hemos creado en el pre procesado y que se le pasa al workflow en el momento de invocarle. Ese fichero contendrá el valor de esa variable “\$InputFile\_BD”.

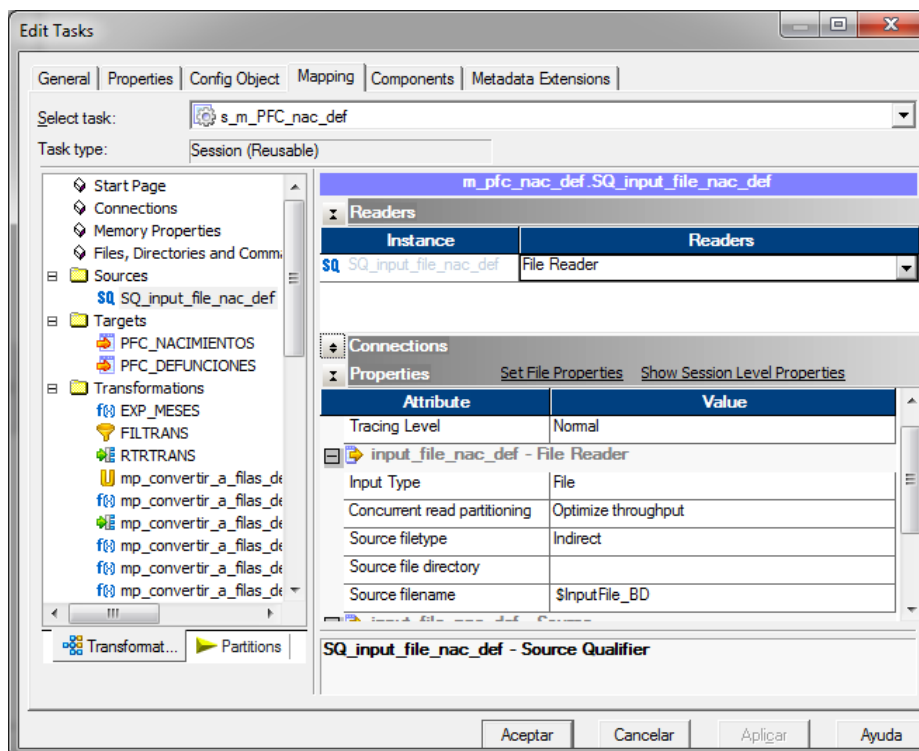


Ilustración 149: Configuración de orígenes en la sesión.

Con respecto a la configuración de los destinos en primer lugar al igual que en los orígenes tendremos que identificar si la carga la realizaremos en un fichero o en una tabla de bbdd. En este caso dado que cargaremos los datos en bbdd elegimos la opción “RelationalWriter” y acto seguido tendremos que indicarle que conexión a bbdd usaremos de todas las que podamos tener configuradas.

Por último, en el apartado de “Properties” de la pestaña Mapping tendremos que indicar en caso de que nuestro destino sea una tabla de bbdd, qué tipo de carga en destino queremos realizar, indicando en el campo “Target load type” entre las opciones “Normal” que realizaría el commit de los registros según los inserta en la tabla y la otra opción podría ser elegir “bulk” con la cual solo se hace commit al final de la carga de todos los registros, lo que implica que si hubiera algún fallo que parara la carga no se habrían cargado ninguno de los datos.

Además podremos, como en este caso, indicarle a través del campo “Rejectfilename” que los registros descartados en la carga por el motivo que fueran se guarden en un fichero para su posterior estudio.

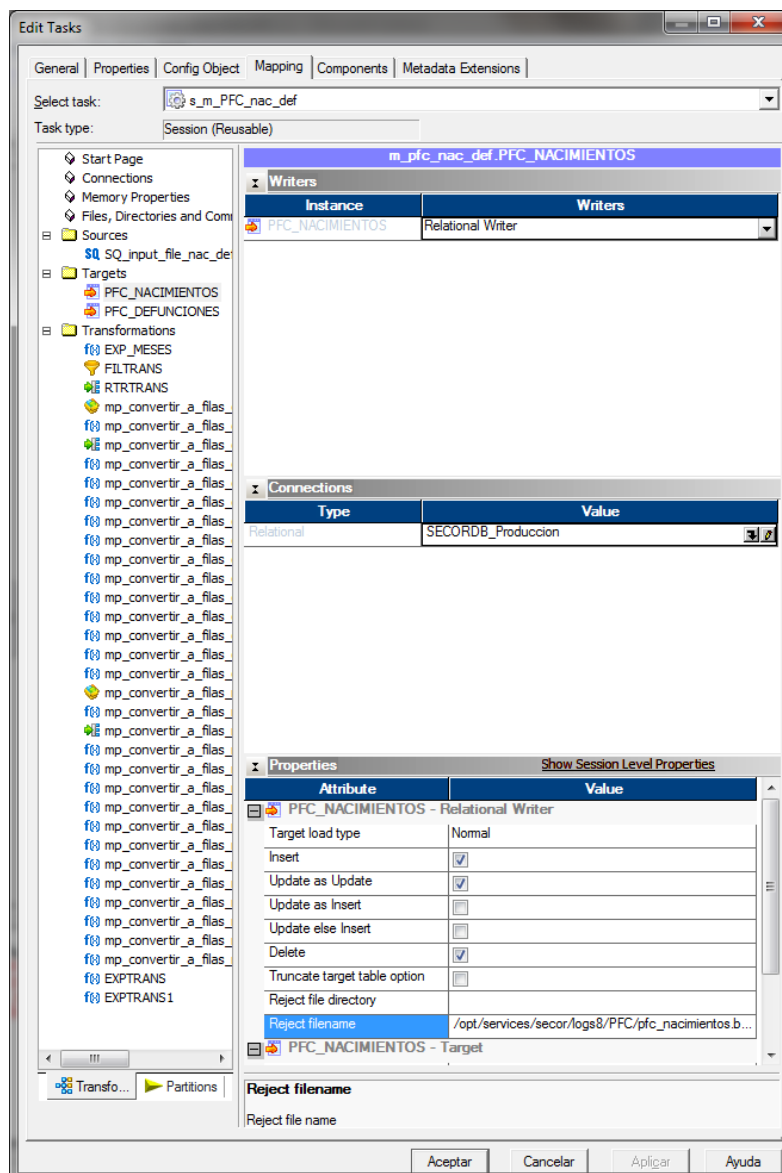


Ilustración 150: Configuración de destinos en la sesión.

### 3.4.2 Carga de datos para nacimientos y defunciones nacionales extranjeros.

En este proceso de carga realizaremos la carga de los datos de nacimientos con padres o madres extranjeras y defunciones de personas extranjeras que residen en España atendiendo al mes y año del suceso, la provincia y al sexo.

En primer lugar, realizaremos al igual que en el proceso anterior una lectura de los ficheros de origen que tendrán la estructura indicada en el apartado [1.3 Estructura de las fuentes](#). En este caso el fichero de entrada tendrá los datos de hombres y mujeres en distintos campos del mismo registro. La lectura del fichero de entrada tendría el siguiente aspecto.

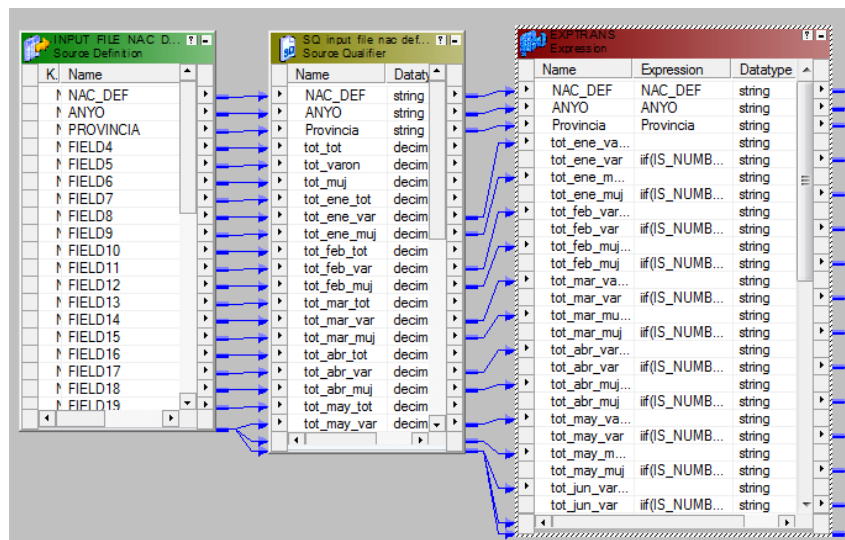


Ilustración 151: Lectura del fichero de entrada.

Como se puede ver una vez leído un registro del fichero se recibe como entrada en una expresión, dicha expresión realiza la asignación de los valores por meses y sexo a cada uno de los campos y a su vez realiza la transformación de tipos pasamos de una cadena de texto (String) que recibimos como entrada a valores numéricos que depositamos en la salida.

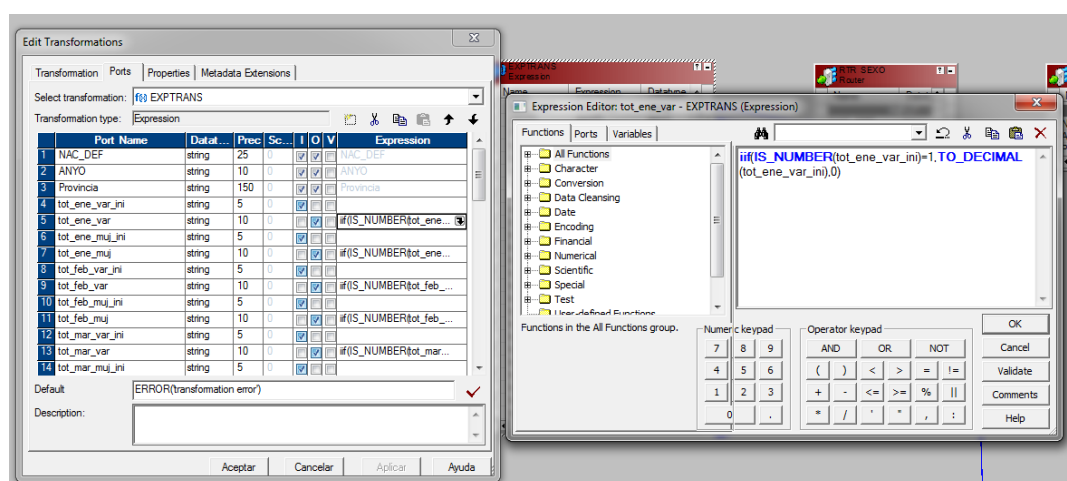


Ilustración 152: Configuración de la expresión y función de cambio de tipos.

Como se muestra en esta ilustración podemos ver la configuración de la expresión y como se asignan los campos de entrada a las distintas salidas ya separados por sexo y mes. La transformación que se realiza es la siguiente:

**iif(IS\_NUMBER(tot\_ene\_var\_ini)=1,TO\_DECIMAL(tot\_ene\_var\_ini),0)**

Comprobamos si el valor que tiene el campo tot\_ene\_var\_ini (total de defunciones o nacimientos de varones en enero) es un numero, esto dará como resultado 1 si es un numero y 0 si no lo es. En caso afirmativo transformaremos ese campo al tipo decimal mediante la función to\_decimal y en caso de no ser un numero le pondremos por defecto el valor 0 ya que si no se ha producido una muerte o nacimiento en un mes en concreto ese campo vendrá a vacío en el fichero de entrada.

Una vez realizadas estas transformaciones, lo que haremos será diferenciar ya los datos que corresponderán a varones y a mujeres. Como la información para varones y mujeres viene en el mismo registro pero los campos vienen diferenciados y sabemos cual es cada campo necesitaremos realizar en este punto un primer producto cartesiano sin necesidad de diferenciar en la condición del router el sexo ya que como digo sabemos que datos queremos obtener para cada sexo por el orden de los campos y duplicar los registros iniciales unos con los datos de los hombres y otros con los datos de las mujeres. Para realizar esto utilizamos un router sin ponerle condición de filtro en los grupos y sacando en cada grupo solo los campos propios de este grupo.

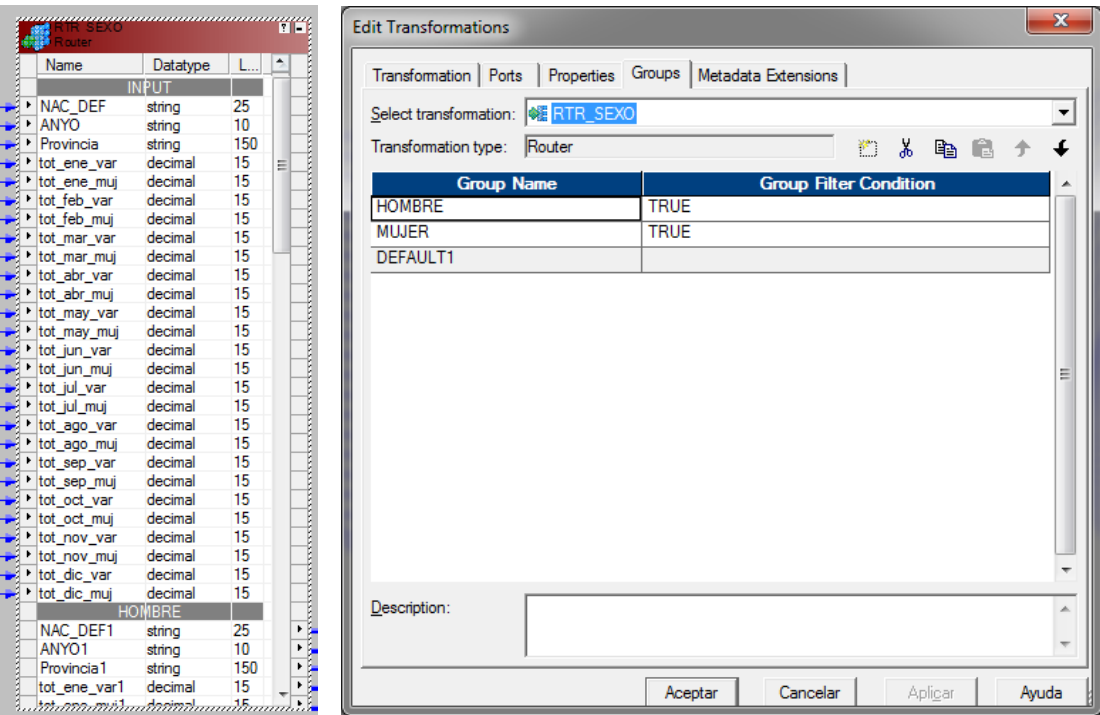


Ilustración 153: Configuración del router.

Llegados a este punto ahora realizaremos en el mapeo principal lo mismo que realizamos en el proceso anterior de nacimientos y defunciones nacionales con el mapplet. De esta manera podremos ver como el uso de los mapplets ayuda a no tener que duplicar código y simplifica mucho el mantenimiento y comprensión de los procesos de carga. Ahora vamos a realizar mediante dos routers uno para los hombres y otro para las mujeres, un producto cartesiano de los registros con el fin de conseguir trasponer los registros actuales que contienen toda la información de las mujeres y los hombres de todos los meses del año en un unico registro en 12 registros mensuales atendiendo al sexo.

Como se muestra en la imagen siguiente, una vez más conseguimos el cartesiano deseado no indicándole ninguna condición de filtrado a los frupos creados y sacando en cada més sólo los datos del mes correspondiente, datos que por el orden de los registros ya sabemos cuales son los que corresponden a cada mes y sexo.

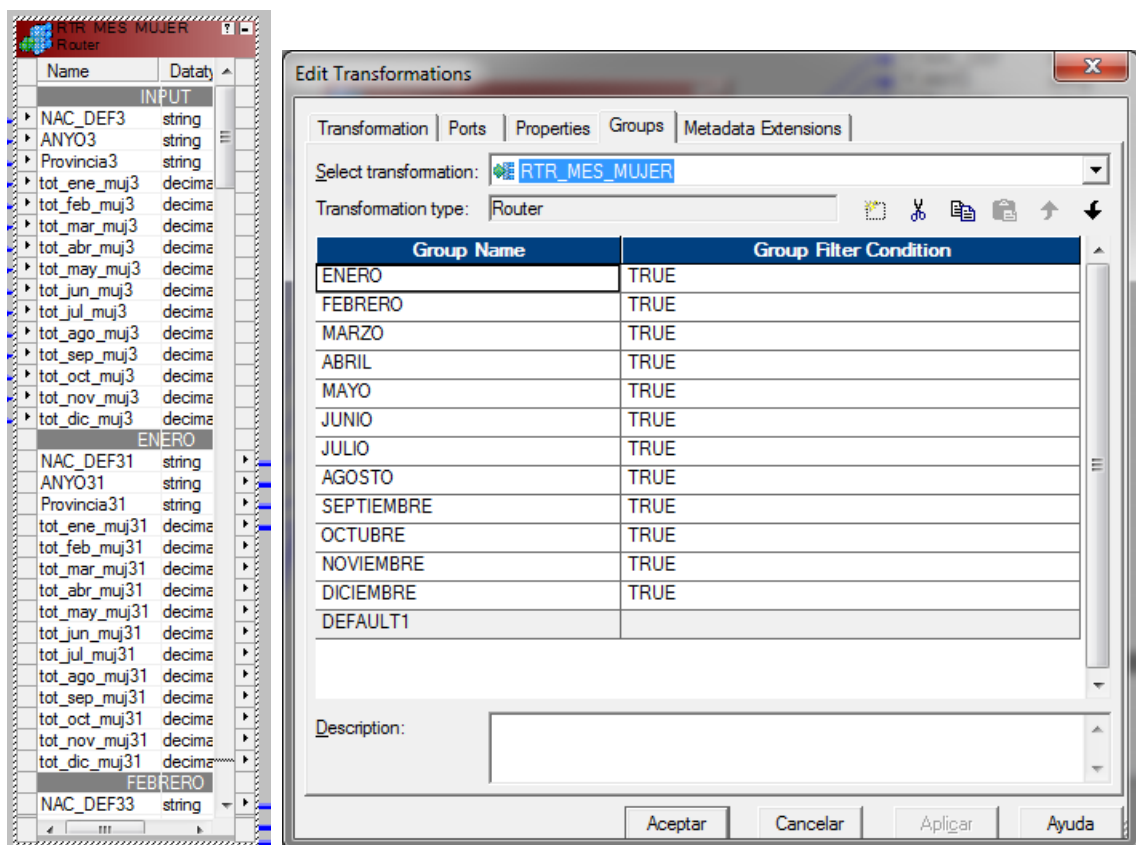


Ilustración 154: Router para la bifurcación mensual para mujeres.

Una vez realizado el cartesiano, del router salen los datos mensuales según el sexo hacia una expresión, habra una expresión para recoger los datos para cada mes y sexo del individuo. En estas expresiones se calculan las fechas, una vez más la fecha final se compone del mes en inglés más los dos últimos caracteres del año, que obtendremos realizando un substr del año. También se asigna el valor del sexo que tenemos como parametros declarados y que se pasan desde el fichero de parametros en el momento de la invocacion al workflow desde el shell script.

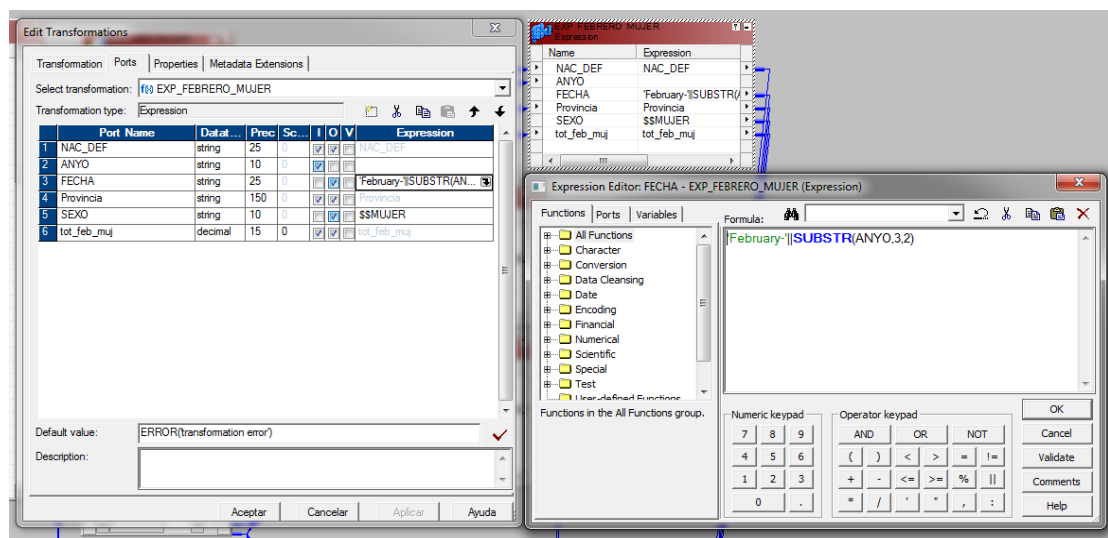


Ilustración 155: Configuración de la expresión mensual y expresión para la fecha.



Como se puede apreciar en esta imagen el campo “SEXO” viene establecido por el parámetro \$\$MUJER igualmente hay otro similar en las expresiones mensuales para los datos de hombres \$\$HOMBRE, estos dos parámetros tienen los valores “Hombre” y “Mujer” que se definen en la siguiente pantalla.

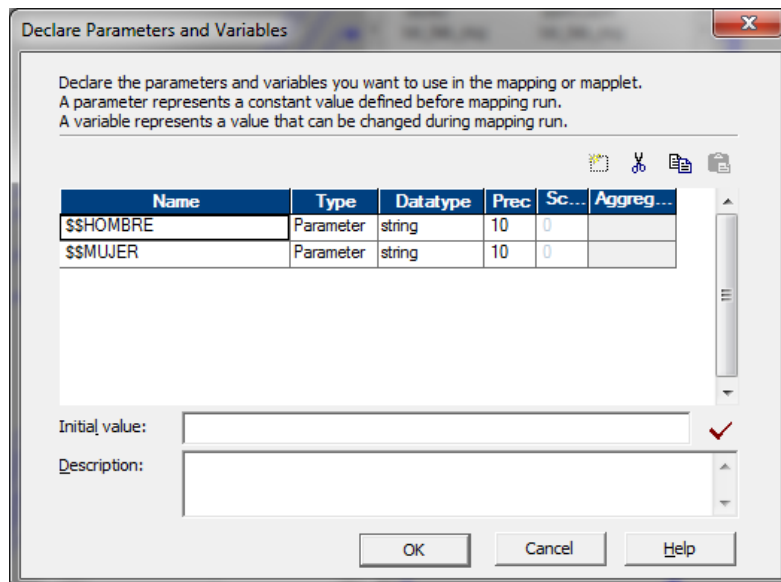


Ilustración 156: Definición de parámetros de entrada.

Una vez que hemos realizado el cartesiano de los meses para el sexo correspondiente realizaremos la unión de estos registros de nuevo en un único flujo con un operador UNION tal cual hicimos en el proceso de carga anterior.

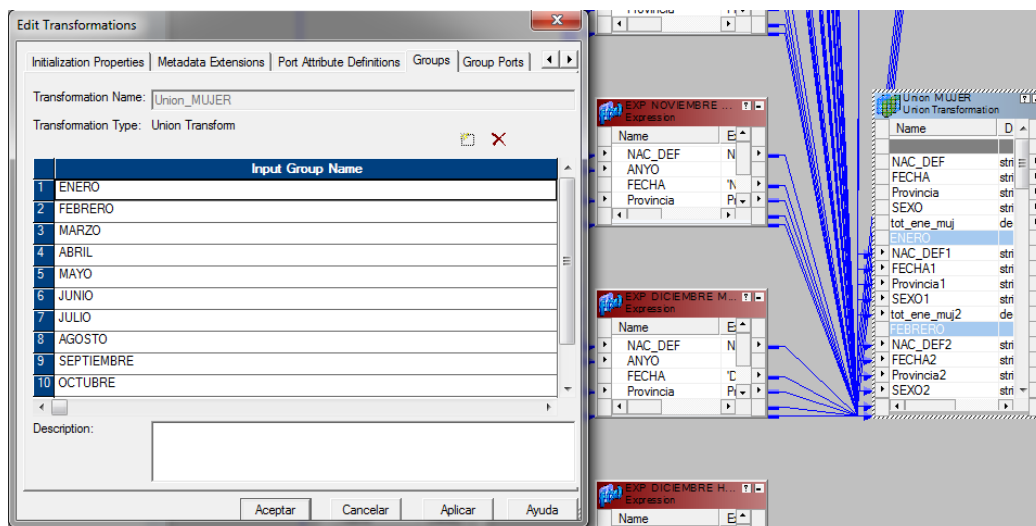


Ilustración 157: unión de los meses en el componente UNION y configuración del mismo.

Ahora volvemos a unificar los flujos de los sexos y tendremos como resultado un único flujo que contendrá los registros de nacimientos y defunciones por sexo y mes separados por esta clave valor.

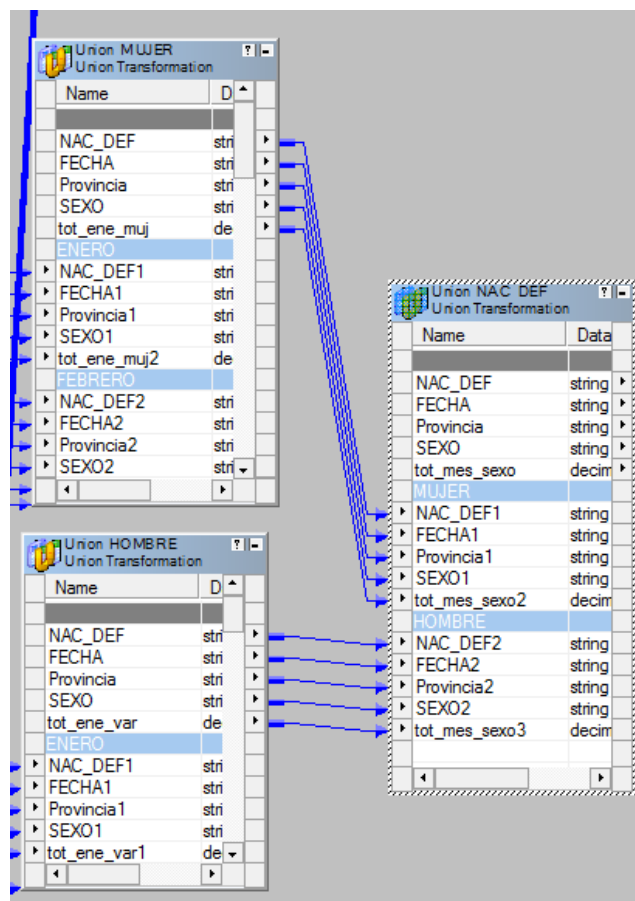


Ilustración 158: Unificación de flujos por sexo.

Finalmente realizaremos una última bifurcación para diferenciar los nacimientos y las defunciones para su posterior inserción en la bbdd.

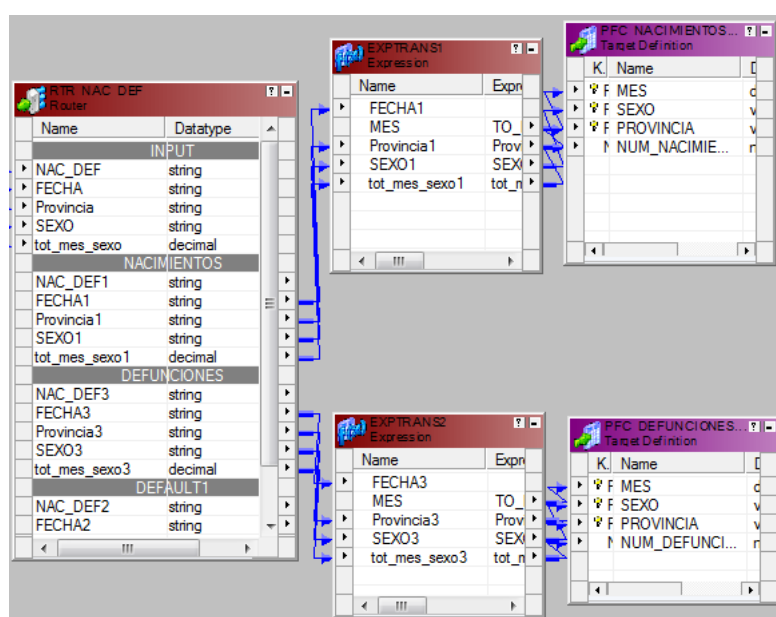
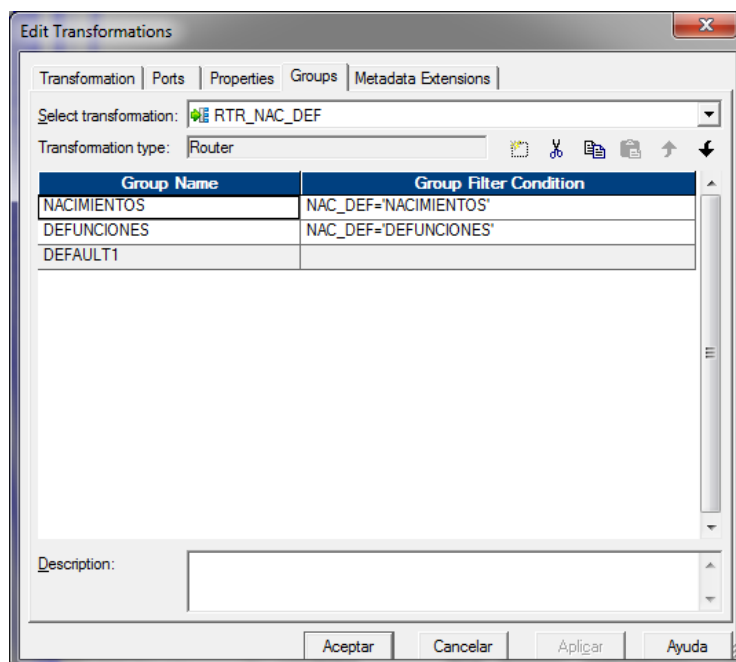


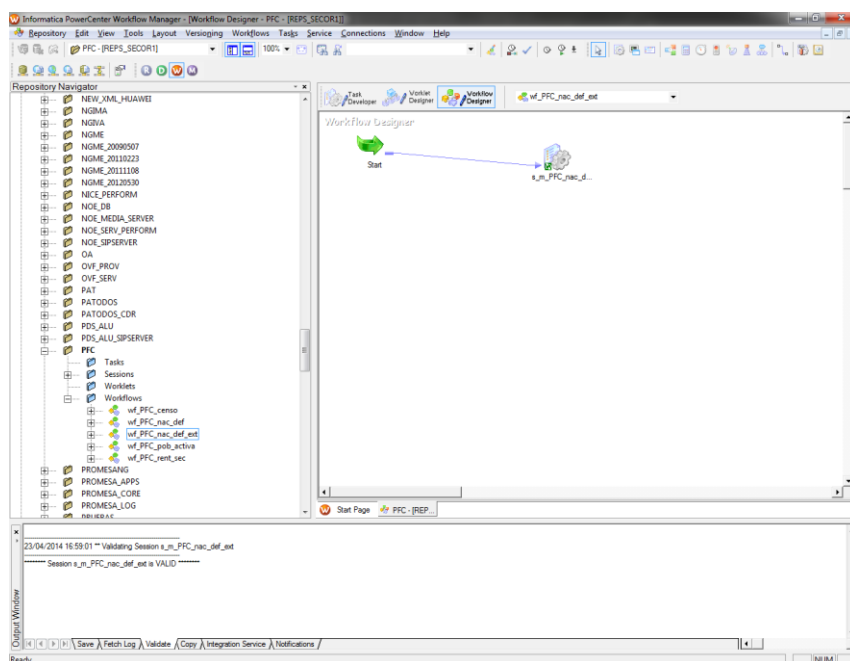
Ilustración 159: Bifurcación por nacimiento o defunción e inserción en tabla.

Para bifurcar el flujo antes de la inserción en la tabla correspondiente de bbdd se configura el componente router de la siguiente manera, se verifica el valor entrante del campo NAC\_DEF y lo re direcciona por un grupo dependiendo de si el valor es “NACIMIENTOS” o “DEFUNCIONES”.



**Ilustración 160: Configuración del router de nacimientos y defunciones.**

A nivel de workflow, en este caso es igual que ocurre con el proceso de nacimientos y defunciones nacionales. Tenemos un único workflow con una única sesión que se configura igual que la sesión de lo nacimientos nacionales.



**Ilustración 161: Vista general del workflow.**

### 3.4.3 Carga de datos para el censo poblacional.

Este proceso de carga, es una carga sencilla y casi directa, el mapeo no sufre ningún cambio de flujo principal y solo sufre una bifurcación a la hora de diferencias los registros que corresponden a un tamaño de municipio u otro.

Aquí se muestra una imagen general del mapeo.

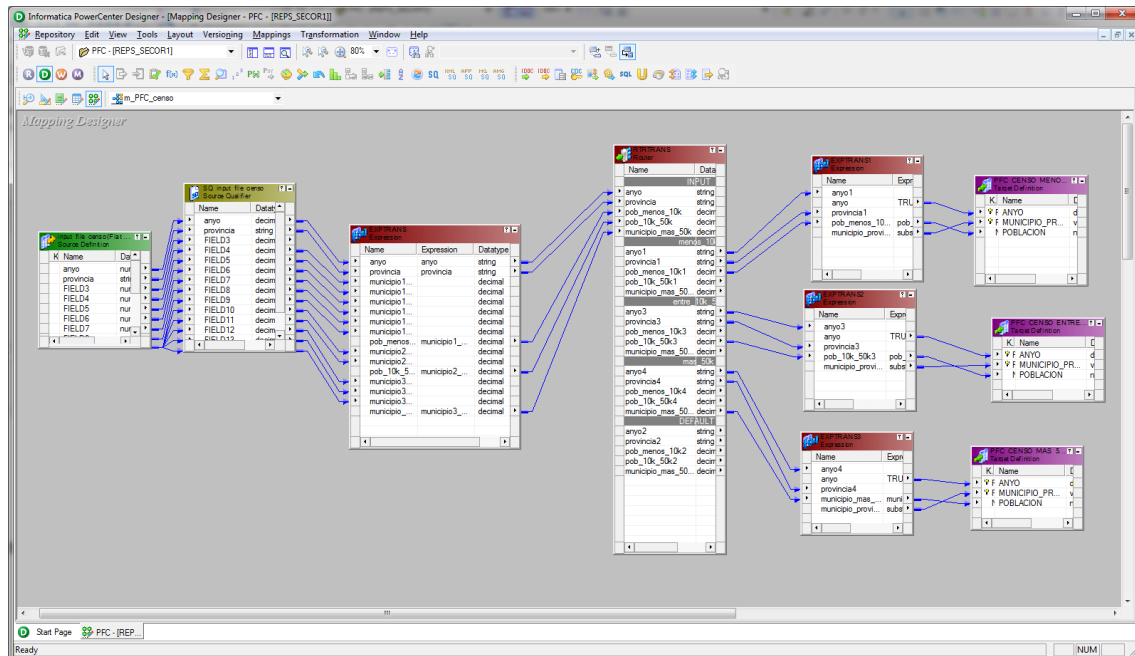


Ilustración 162: Vista general del mapping de censo poblacional.

En un primer momento tras la lectura del fichero de entrada, los datos llegan hasta una expresión en la cual se sumaran los valores de varios campos para unificarlos finalmente en tres nuevos campos sumatorios. Esto es debido a que un registro de entrada tiene más divisiones de poblaciones por municipio de las que se están tratando a lo largo de todo el proyecto que son tres: “municipios con menos de 10000 habitantes”, “municipios con entre 10001 y 50000 habitantes”, “municipios con más de 50000 habitantes”. Sin embargo este fichero de entrada tiene 11 divisiones de municipios que englobaremos de la siguiente manera: los seis primeros valores de población se suman para formar el valor definitivo de la población en municipio de menos de 10000 habitantes, los dos siguientes campos suman la población de los municipios de entre 10001 y 50000 habitantes y por último los tres últimos campos suman la población de los municipios de mas de 50000 habitantes.

La fórmula que se usa para realizar dichas sumas es la siguiente:

**“municipio1\_p1 + municipio1\_p2 + municipio1\_p3 + municipio1\_p4 + municipio1\_p5 + municipio1\_p6”**

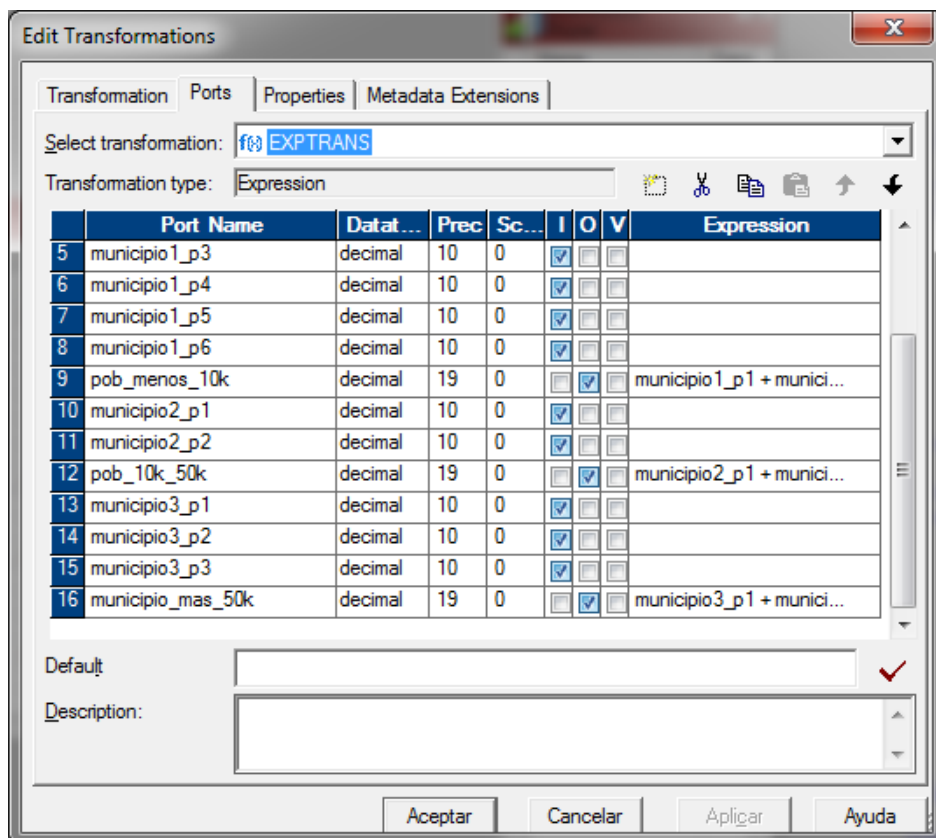


Ilustración 163: Configuración de la expresión de suma de poblaciones de municipios.

Una vez realizada esta agrupación de campos y obteniendo solo los campos de salida que vamos a llevar hasta el final del proceso de carga, pasamos el registro a un router para realizar un producto cartesiano y obtener por cada registro de entrada tres registros con la clave elegida (año, provincia) y un único valor para el tipo de municipio que queremos de los tres anteriormente citados.

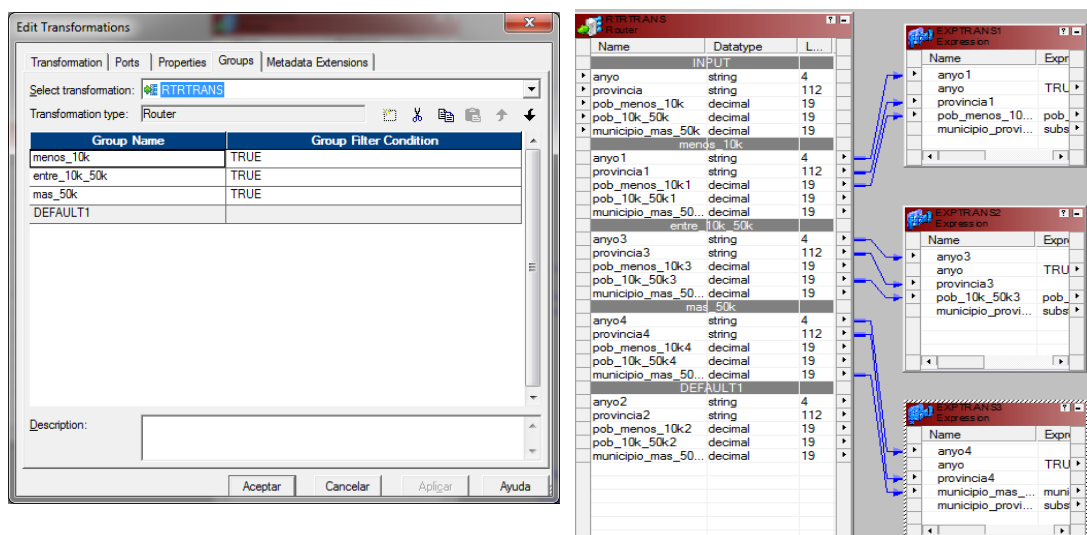


Ilustración 164: Router y configuración de los grupos del mismo.

Una vez realizado el producto cartesiano que nos dividirá el flujo en tres nuevos flujos con un registro cada uno por rango de municipios, los datos llegan a sendas expresiones en las que se modificará el formato del año que recibimos y compondremos el campo municipio\_provincia, además de continuar con el valor numérico de la población para ese municipio que ya habíamos calculado previamente.

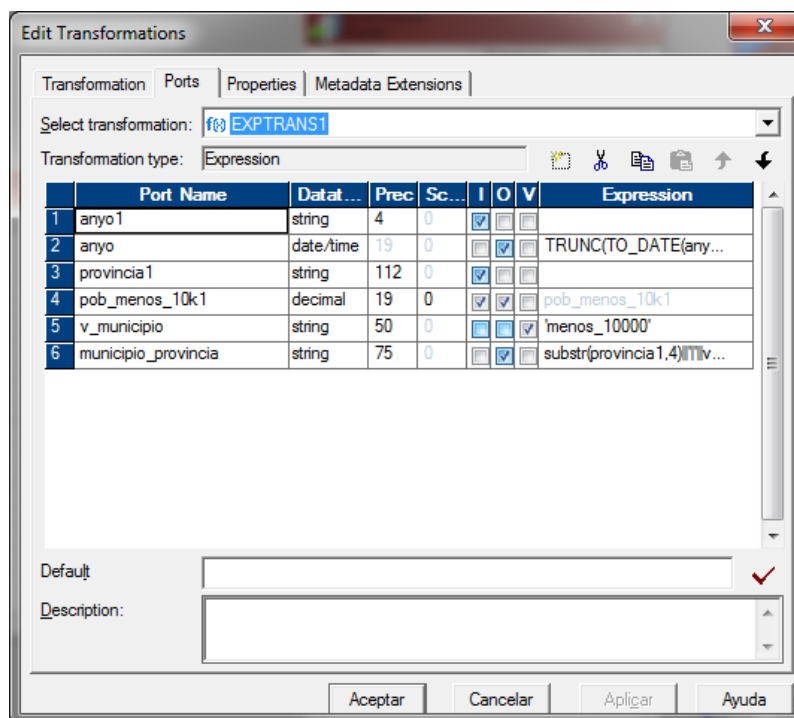


Ilustración 165: Configuración de las expresiones de censos por municipios.

Para calcular el año con el nuevo formato se aplica la siguiente formula:

**TRUNC(TO\_DATE(anyo1, 'yyyy'), 'yyyy')**

En ella se transforma el año de entrada en tipo fecha dándole la máscara de año y además truncamos la fecha para solo quedarnos con el año.

Además creamos para cada opción del router una variable que contendrá el literal correspondiente a cada tipo de municipio: "menos\_10000", "entre\_10001\_50000" y "mas\_50000".

Por último creamos el campo MUNICIPIO\_PROVINCIA con la siguiente función:

**substr(provincia1, 4) || '|' || v\_municipio**

Aquí se indica que el contenido que venga en el campo provincia se cogerá la cadena que comprenda desde el cuarto carácter hasta el final de la palabra y eso será la provincia (esto se hacia así, porque si recordamos los ficheros de entrada venían todas las provincias con un código numero previo mas un guion bajo y de esta manera solo nos quedamos con la provincia). Además le concatenamos la variable V\_MUNICIPIO que habíamos calculado previamente.

Una vez tenemos estas expresiones para cada tipo de municipio preparadas podremos insertar ya directamente en las tablas de destino.

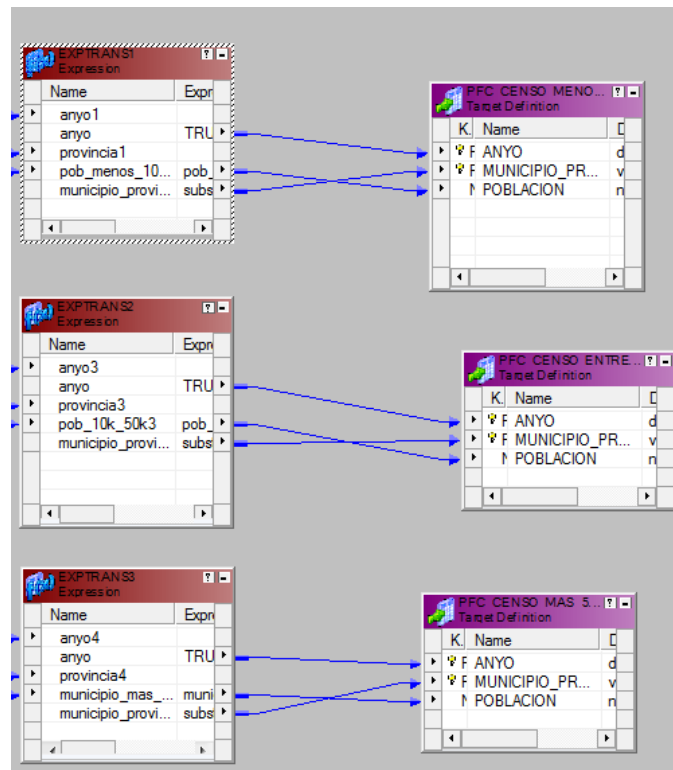


Ilustración 166: Inserción en las tablas de destino.

Como se puede ver aquí se han creado tres tablas de destino, pero realmente son la misma tabla "PFC\_CENSO", aunque es referenciada por decirlo de alguna manera por cada uno de los tres distintos flujos de datos que se refieren a un tipo de municipio.

A nivel de workflow este proceso es igual que los demás. Es un workflow simple con una única sesión.

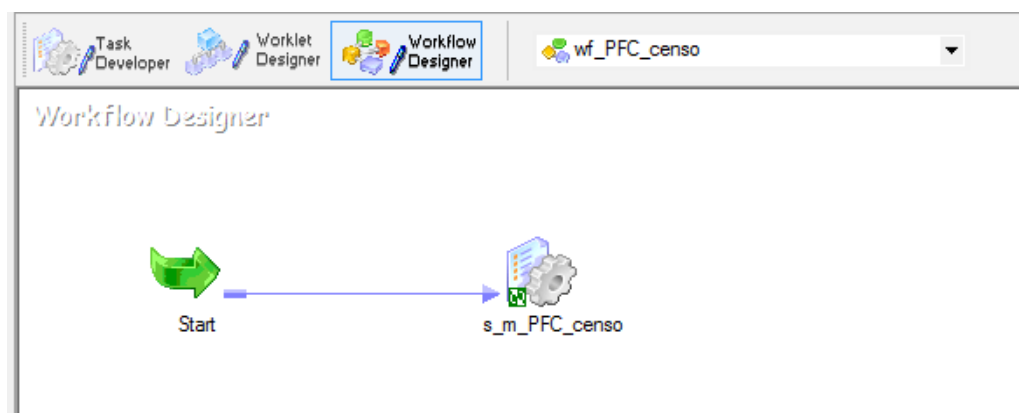


Ilustración 167: Workflow PFC\_CENSO.

Esta sesión tiene configurado un fichero de entrada de tipo indirecta y parametrizado el nombre del fichero que contendrá las referencias a los ficheros a procesar.

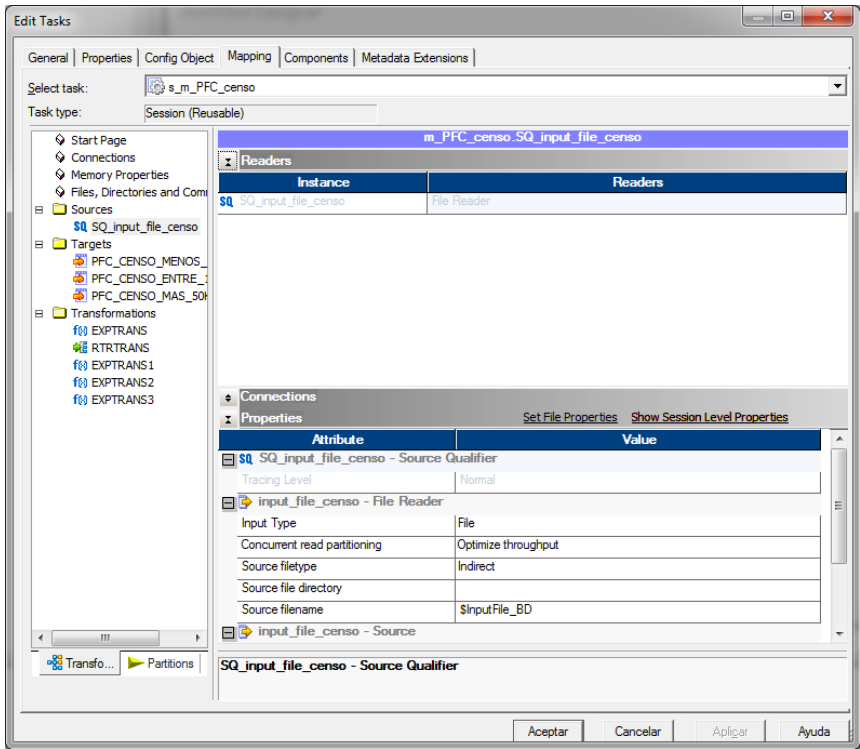


Ilustración 168: Configuración del fichero de entrada.

De igual manera en lo referente a las salidas tenemos creadas tres salidas que referencias a la misma tabla y estas se configuran indicando que se trata de una tabla de bbdd, a que bbdd nos vamos a conectar y en las propiedades propias de la salida indicaremos que queremos que todas usen un tipo de carga normal (según llega un registro realizar el commit) y en que directorio y fichero queremos que guarden los registros rechazados.



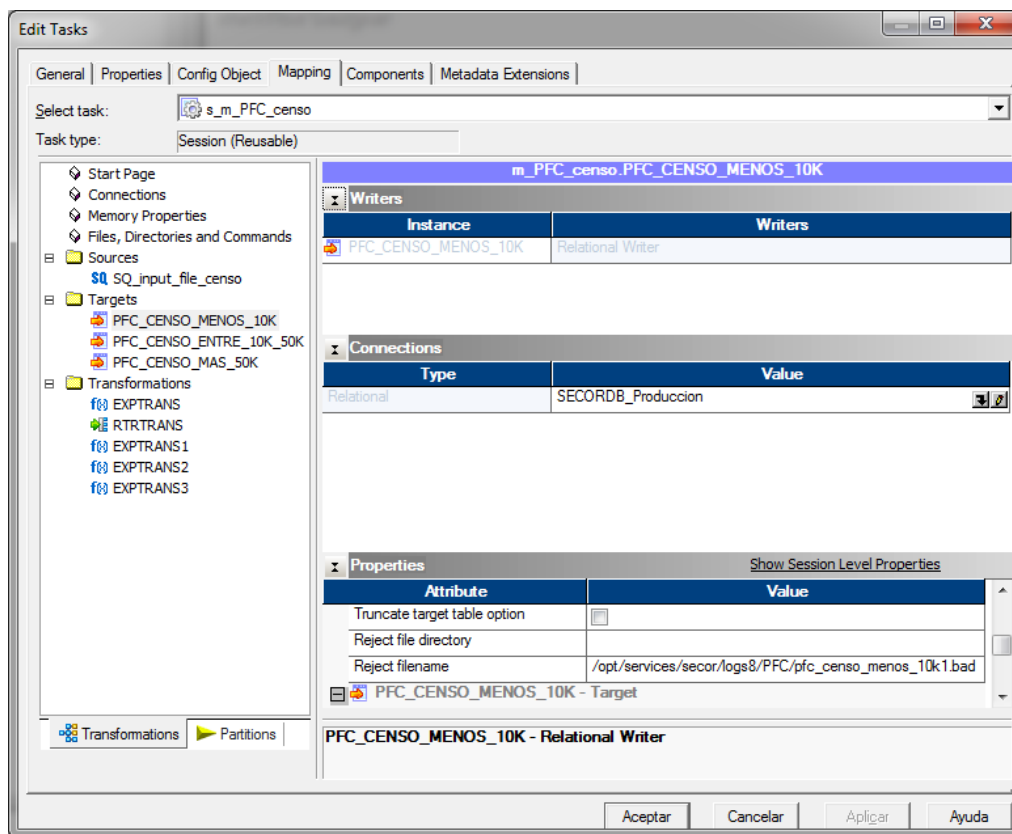


Ilustración 169: Configuración de las tablas de salida.

#### 3.4.4 Carga de datos para rentas por sector.

Este proceso de carga procesará los datos de las rentas por comunidad, sector y sexo durante los años 2004 al 2011. Tiene la particularidad de que estos datos vienen agrupados en dos ficheros de 4 años cada uno. Por ello hemos tenido que insertar un campo adicional el que se indique que fichero se esta tratando, si los datos del periodo 2004 al 2007 o si son los del periodo 2008 al 2011. Además los datos de cada año vienen en distintas columnas por lo que gracias a que sabemos este orden hemos podido una vez que estamos procesando los datos saber que dato se corresponde con cada año en cada fichero.

La vista general del extractor es la siguiente:

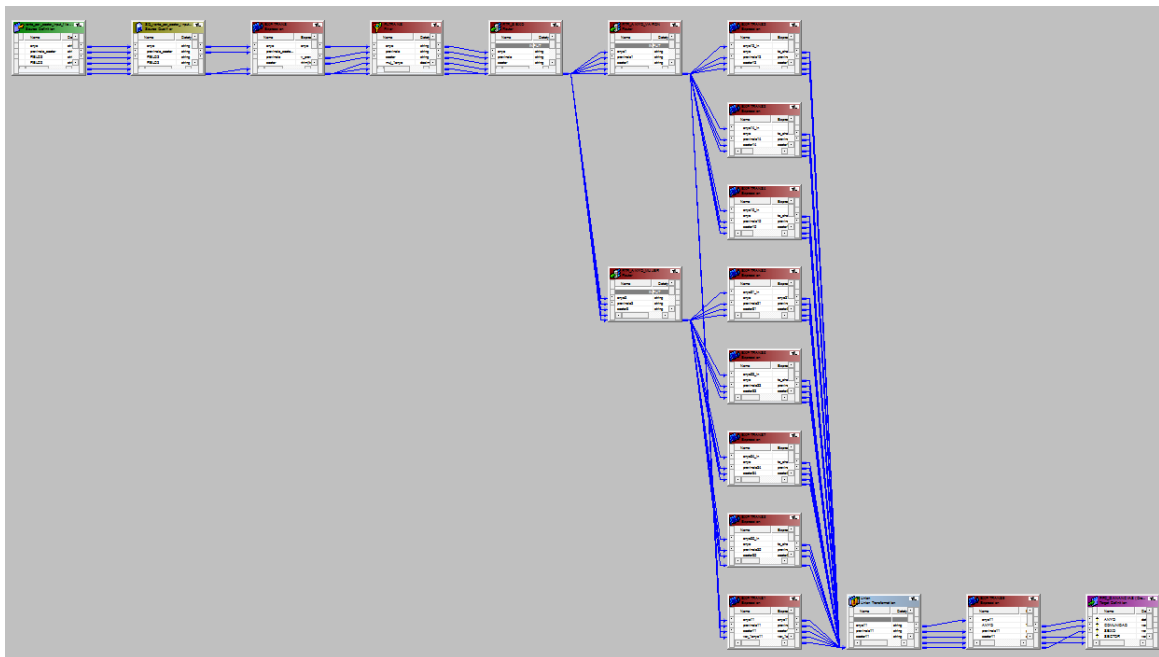


Ilustración 170: Vista general del mapping PFC\_RENT\_SEC.

En un primer momento procedemos a realizar una lectura del fichero de origen y recibir estos datos en una expresión inicial en la cual con la ayuda de dos variables obtendremos la provincia y el sector del cual serán los datos de ese registro, de igual manera realizaremos el cambio de tipo de datos para los valores de las métricas y cambiaremos el tipo cadena de texto inicial (String) por un campo tipo numérico (decimal).

En este caso la configuración de la expresión será la siguiente:

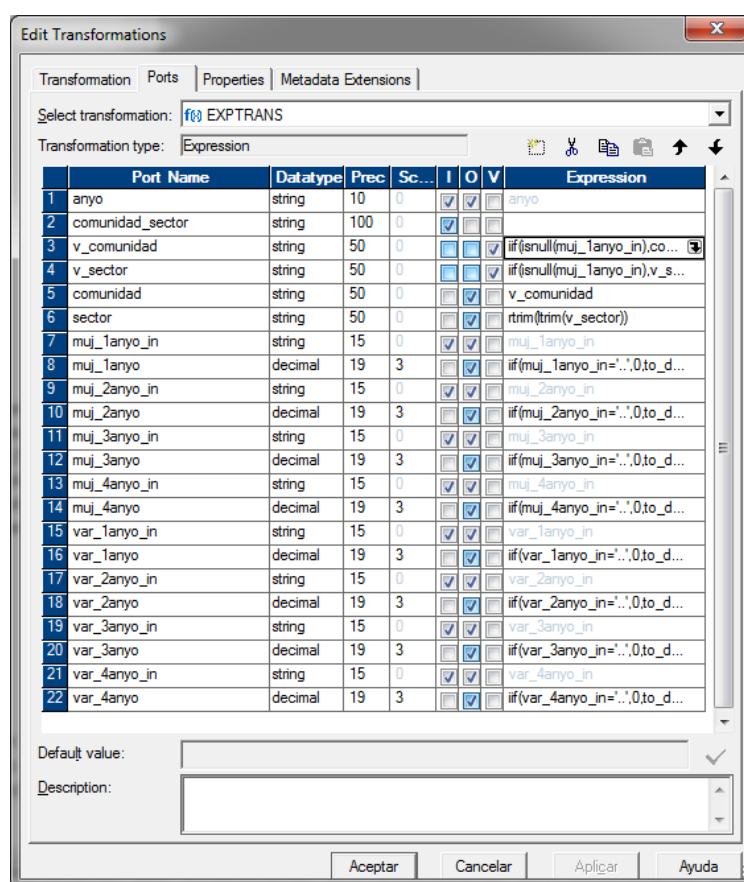


Ilustración 171: Expresión inicial, cambio de tipos y calculo de provincia y sector.

Ahora vamos a ver como se obtiene el valor de cada variable y del resto de campos:

**V\_COMUNIDAD:** Se realiza la siguiente operación **“iif(isnull(muj\_1anyo\_in),comunidad\_sector,v\_comunidad)”** en ella se comprueba si el campo que contendría los valores de la renta para las mujeres del primer año del registro es nulo esto significaría que estamos ante un registro que solo contiene el nombre de la comunidad, por lo que en esta variable guardaríamos el valor del campo **comunidad\_sector** que contiene el nombre de la comunidad. De no ser nulo el primer valor significaría que estamos ante un registro de métricas por lo que en esta variable mantendríamos el valor que tuviéramos guardados previamente.

**V\_SECTOR:** Para la obtención de este campo calculado se aplica la siguiente fórmula: **“iif(isnull(muj\_1anyo\_in),v\_sector,comunidad\_sector)”**, en ella se comprueba si es null el primer campo de valores que tendría el registro de entrada, de no ser null implicaría que se trata de un registro de datos por lo que el valor del campo **comunidad\_sector** sería el valor del sector al que pertenecen los datos del registro. De esta manera guardamos dicho valor en la variable **v\_sector**, si fuera null se trataría de un registro de comunidad y por lo tanto el campo sector se mantendría con el valor que tuviera antes.

Una vez calculados estos campos eliminamos los posibles espacios en blanco antes y después del valor y los asignamos al campo **comunidad** y **sector**.

Los campos que contienen las métricas de los salarios según el sector y el sexo se obtienen siguiendo el orden que sabemos que tienen en el fichero de origen y además se realiza una transformación de tipos de cadena de texto a números con la siguiente función: `"iif(muj_1anyo_in='..',0,to_decimal(muj_1anyo_in))"`.

Se comprueba si el campo en concreto del dato que vamos a buscar viene a vacío o no, en caso de que venga a vacío se le inicializa a 0 y en caso de que no venga a vacío se realiza la conversión de tipos con la función `to_decimal(string x)` que recibe una cadena de texto como entrada y la transforma a número siempre que esta cadena sea realmente un número.

Una vez que hemos realizado estos cálculos los registros pasan por un operador filtro que realiza el filtrado de los registros que son de datos propiamente dichos y los que son de comunidades, quedándose solo con los de datos, a los que ya hemos añadido el nombre de la comunidad y el sector pertinentes. La condición de filtrado utilizada es la siguiente:

`"notisnull(muj_1anyo) or not isnull(var_1anyo)"`

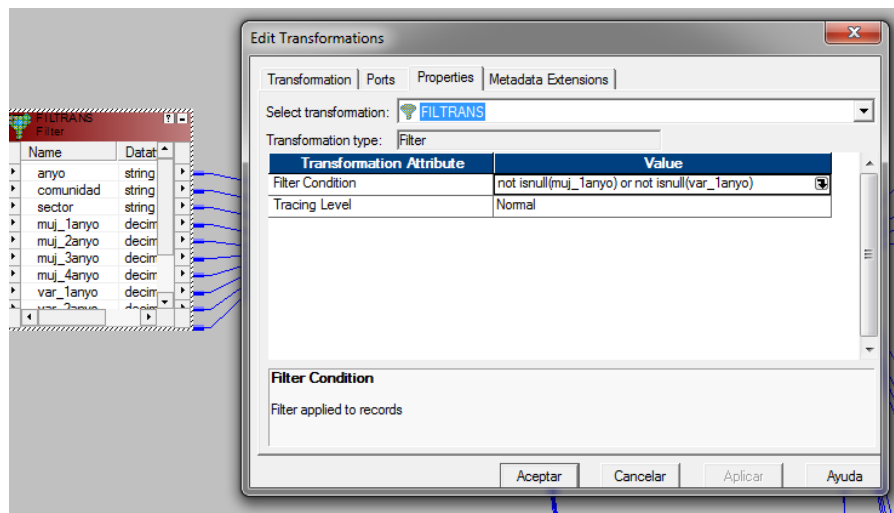


Ilustración 172: Filtro y configuración del filtro.

Posteriormente y una vez que solo tenemos registros de métricas, estos pasan a un componente router que diferenciará los datos según el sexo, realizando un producto cartesiano. Nos llegará un registro que contendrá los datos de los hombres y mujeres juntos y saldrán dos flujos uno con los registros de los hombres y otro con los registros de las mujeres.

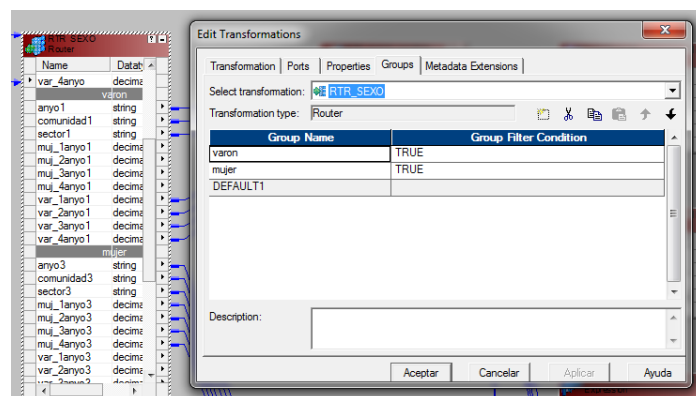


Ilustración 173: Router Sexo y configuración.

A continuación explicaremos uno de los flujos que seguiremos para un sexo en concreto ya que ambos serán exactamente iguales y terminaran uniéndose de nuevo ambos flujos antes de insertarse en bbdd.

Cuando salen los registros por cualquiera de los grupos del router de sexo, tenemos un registro que tiene la estructura de (año, comunidad, sector, valor\_año1, valor\_año2, valor\_año3, valor\_año4) y estos se dirigen a un nuevo router que realizara un producto cartesiano para cada año quedando ahora 4 registros con la estructura (año, comunidad, sector, valor\_añoX). Una vez que salen del router se dirigen cada nuevo registro a una expresión.

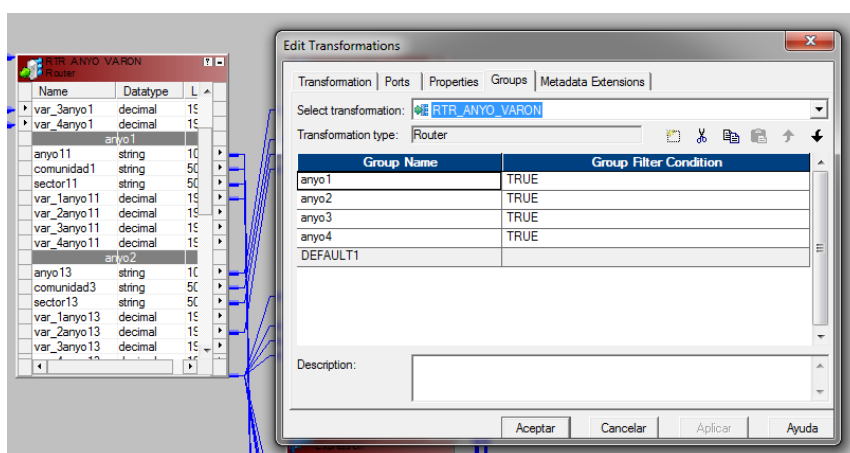


Ilustración 174: Router de años y configuración.

En la expresión que recoge los datos de salida del router solo se añade el campo sexo que es un literal que contendrá los valores “Hombre” o “Mujer”.

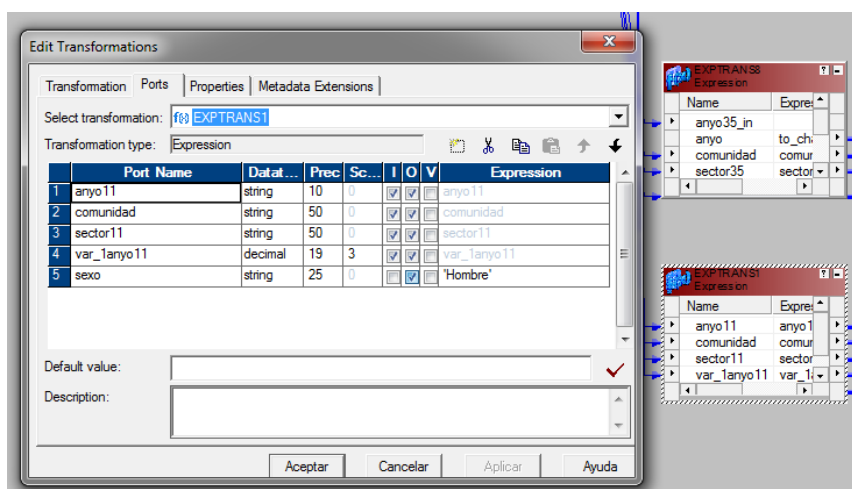


Ilustración 175: Expresión creación campo sexo.

Una vez hemos distinguido cada año y añadido el sexo al que corresponden los datos se vuelve a unificar el flujo a través de un componente UNION, cuyos datos de salida irán a una última expresión que realizará la transformación del año en un campo fecha y que truncará este valor a solo el año con la siguiente formula: “**TRUNC(TO\_DATE(anyo11,'YYYY'),'YYYY')**”.

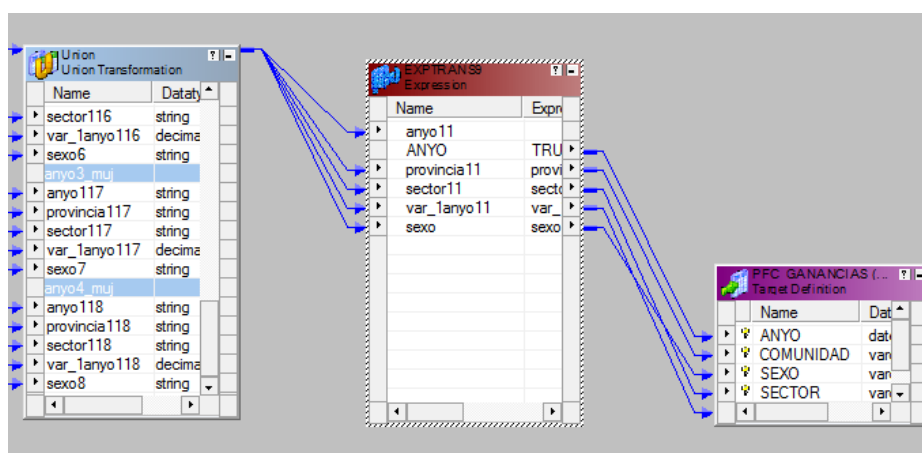


Ilustración 176: Unión de los flujos y carga en destino.

A nivel de workflow se realiza la carga con una única sesión que configuraremos de la siguiente manera.



Ilustración177: workflow PFC\_RENT\_SEC.

En primer lugar configuraremos como siempre el fichero donde queremos que se guarden los logs de ejecución, en nuestro caso en la home de logs tendremos un directorio que se llame PFC y dentro de él se creará el fichero s\_m\_PFC\_rent\_sec.log.

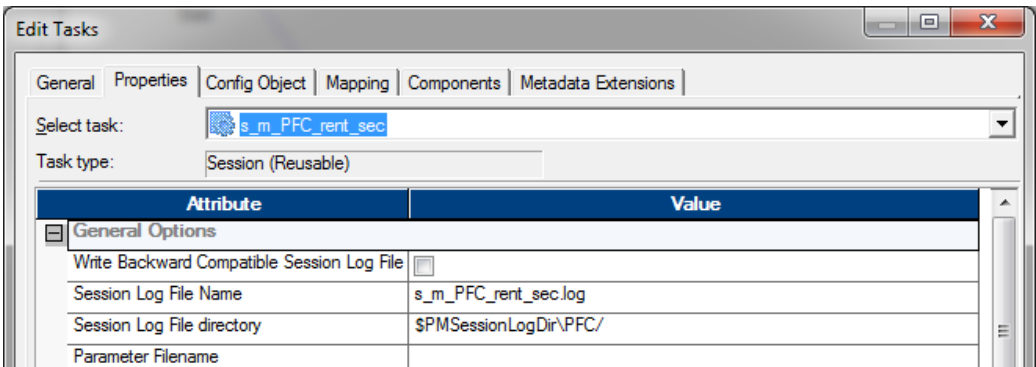


Ilustración 178: Configuración del fichero de logs de sesión.

En el apartado de configuración del mapping, en este proceso de carga tenemos una entrada de fichero pero en diferencia a los procesos creados hasta ahora el fichero es de tipo directo, lo que significa que los datos que leemos del fichero de entrada que recibimos como parámetro en la sesión, contendrá los datos a cargar directamente y no las referencias absolutas a un listado de ficheros que contienen los datos.

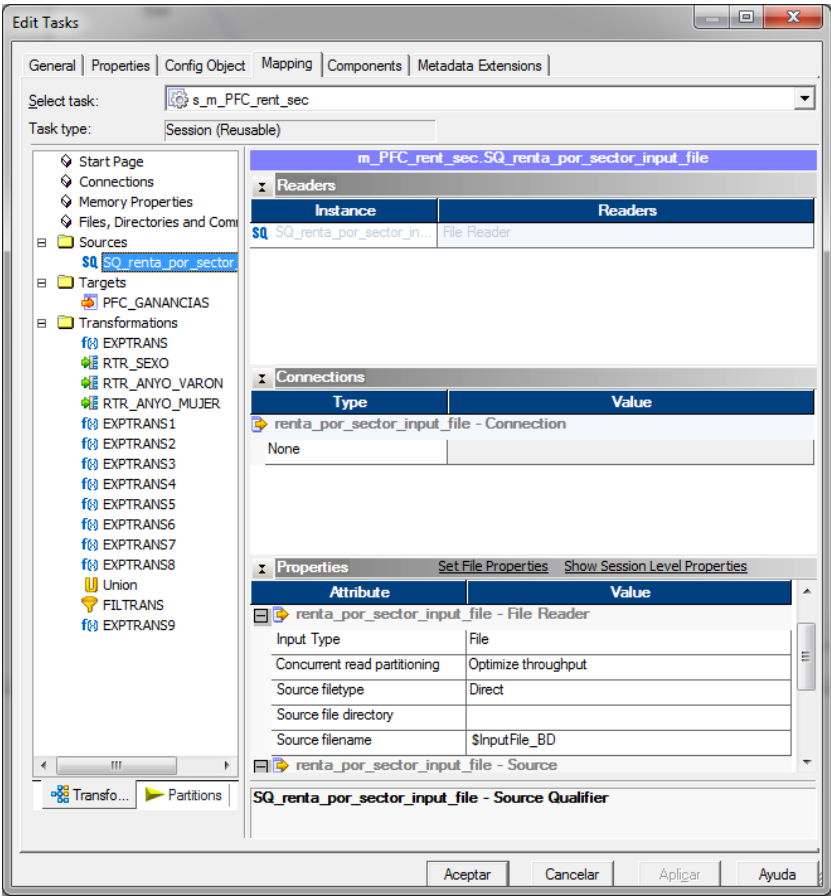


Ilustración 179: Configuración de un fichero de entrada directo.

Con respecto al destino, se configura como hasta ahora indicando que es de tipo relationalwriter, lo que significa que el destino es una tabla de bbdd e indicamos que la bbdd a la que debe conectarse es SECORDB\_Produccion.

Como siempre, indicamos que el tipo de carga queremos que sea normal, lo cual significa que según se inserte un registro en la tabla de destino se realizara el commit del mismo, para esto hay que indicarlo en el campo Target load type. Además en este apartado indicamos que queremos que se guarden los registros rechazados en un fichero en el directorio que nosotros queramos, para esto hay que indicar la ruta y el nombre del fichero en el campo rejectedfilename.

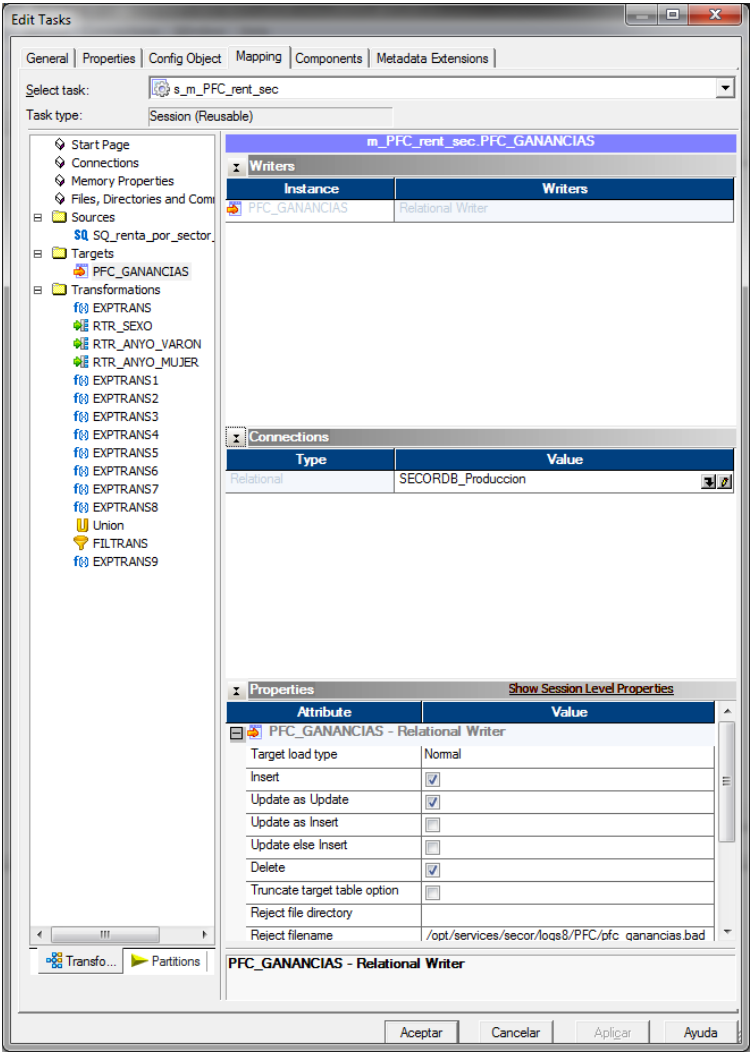


Ilustración 180: Configuración de la tabla de destino PFC\_RENT\_SEC.

### 3.4.5 Carga de datos para la ocupación de la población.

Este proceso de carga es el más completo de los que hemos descrito hasta ahora, realiza la carga de los datos de ocupación laboral de la población atendiendo al tipo de estado (activo\_ocupado, activo\_parado, activo\_parado1empleo, inactivo), año y provincia.



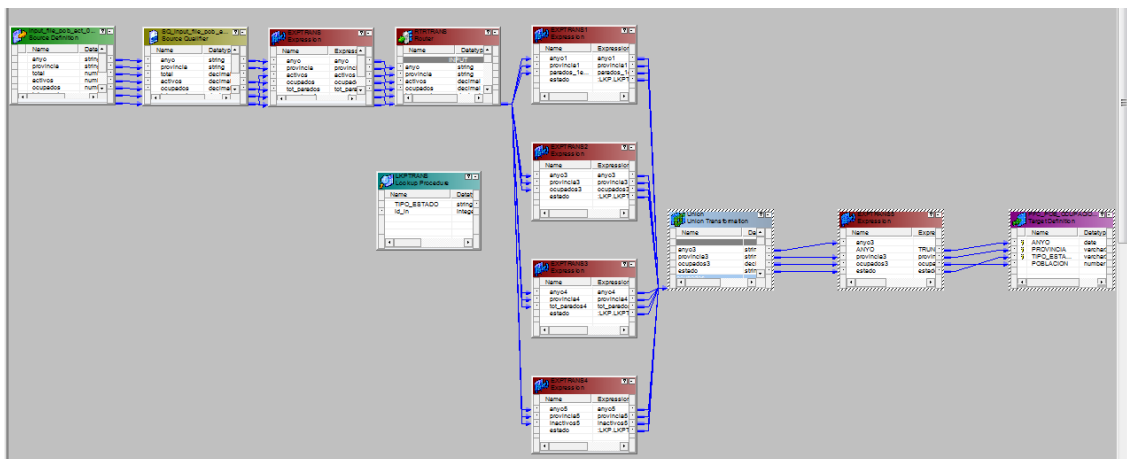


Ilustración 181: Vista general del proceso de carga m\_PFC\_pob\_activa\_04.

Este proceso lo hemos dividido en dos procesos o mapeos distintos para realizar un ejemplo más didáctico y a la par porque vamos a procesar dos ficheros de tipos distintos. Por un lado procesaremos un fichero único para los datos del 2004 y por otro lado procesaremos los ficheros desde el 2005 hasta el 2013, además cada uno de estas dos cargar tendrán ficheros con formatos distintos.

Comenzaremos por explicar el proceso para los datos del 2004, este tiene la estructura que se muestra en la ilustración 52. En un primer momento podemos ver como se realiza una lectura del fichero de origen hasta que estos registros llegan a la primera expresión.

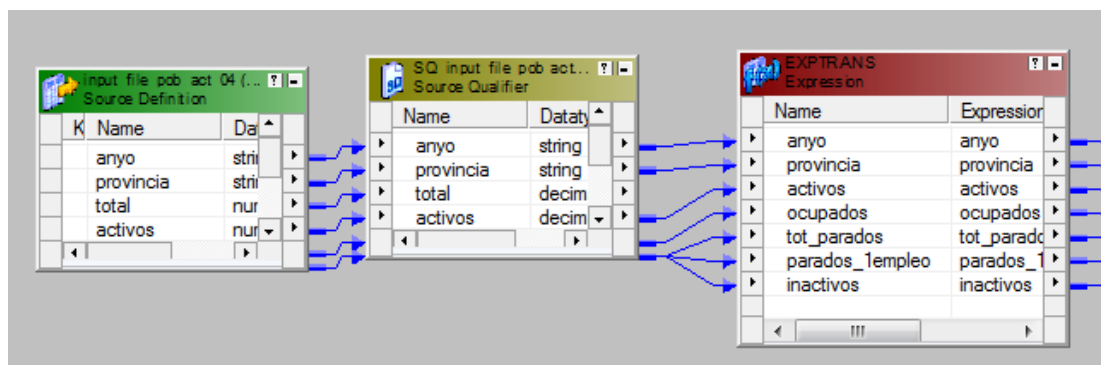


Ilustración 182: Lectura de datos y primera expresión de transformación.

En esta primera expresión no realizamos ningún cambio sobre los campos simplemente los recibimos y los pasamos a un primer router en el cual realizaremos un producto cartesiano que dividirá el registro de entrada que contenía los valores de los cuatro tipos de estado anteriormente mencionados en 4 registros de salida uno para cada tipo de estado.

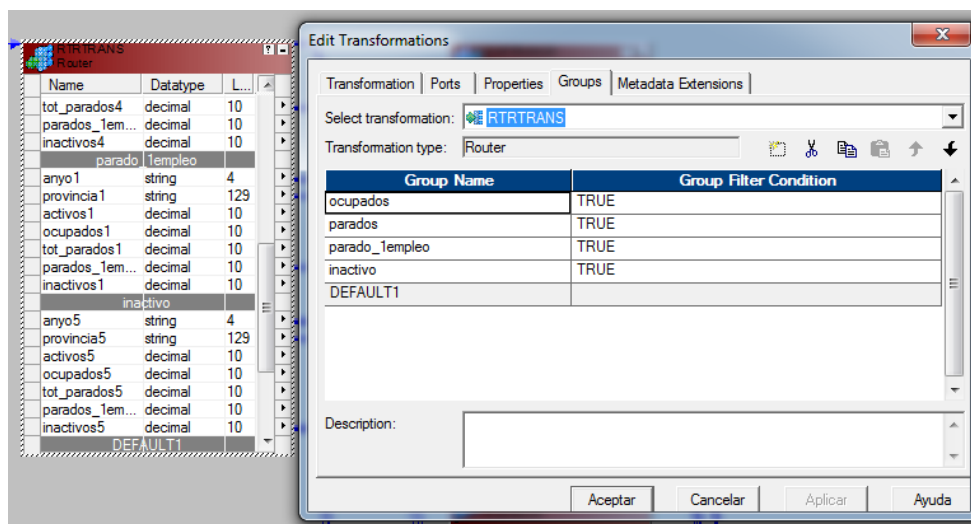


Ilustración 183: Router para tipo de estado.

Una vez que los datos salen del router estos llegan a una expresión en la cual asignaremos para cada tipo\_estado que estamos tratando un valor (posición\_estado) que se cruzará en bbdd mediante una lookup.

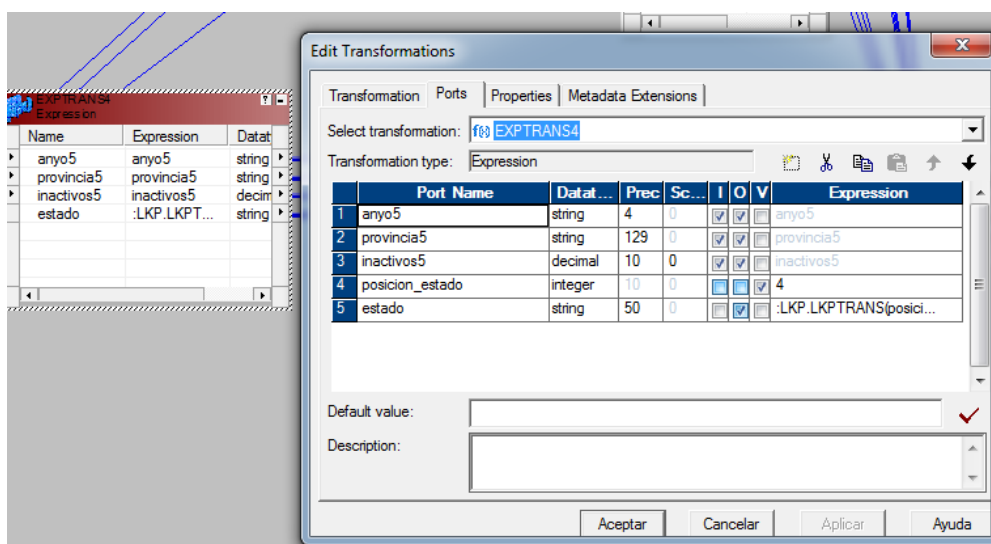


Ilustración 184: Expresión tipo estado y configuración.

Para obtener de qué tipo de estado estamos tratando, hemos asignado a cada tipo de estado en las expresiones un valor del 1 al 4 y mediante una lookup desconectada vamos a realizar una búsqueda sobre la tabla maestra y obtener el tipo de estado correspondiente al valor introducido.

Una lookup es una función que recibe un parámetro, realiza una búsqueda en la tabla sobre la que se esté realizando la lookup y te devuelve los datos que desees. Pueden ser conectadas (se verán en el siguiente proceso de carga) o desconectadas como ésta, en la que podemos realizar la llamada como se muestra en la siguiente ilustración.

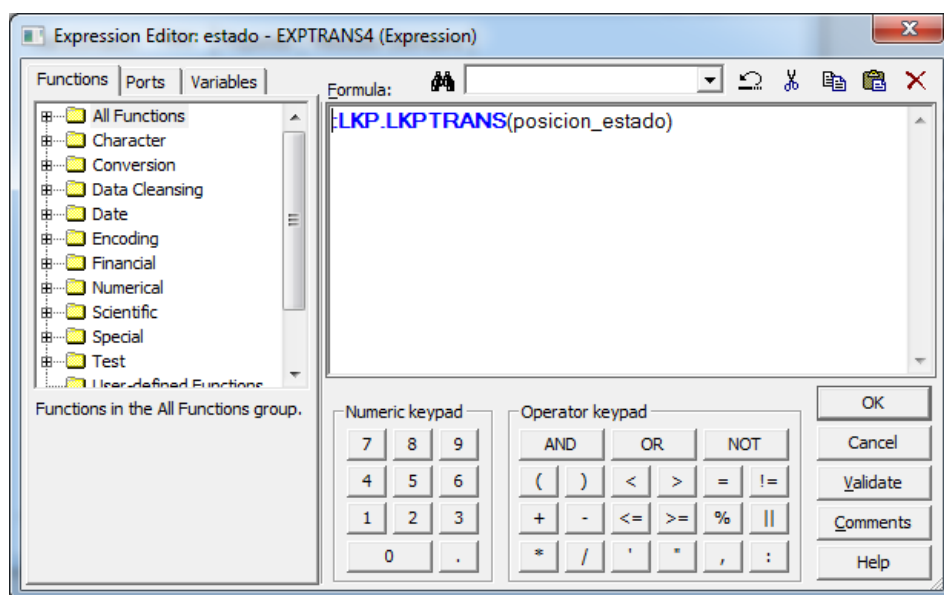


Ilustración 185: Llamada a la lookup.

Como se puede ver se invoca a la lookup y le pasamos como parámetro el campo con el valor que queremos que realice la búsqueda.

La lookup se configura de la siguiente manera:

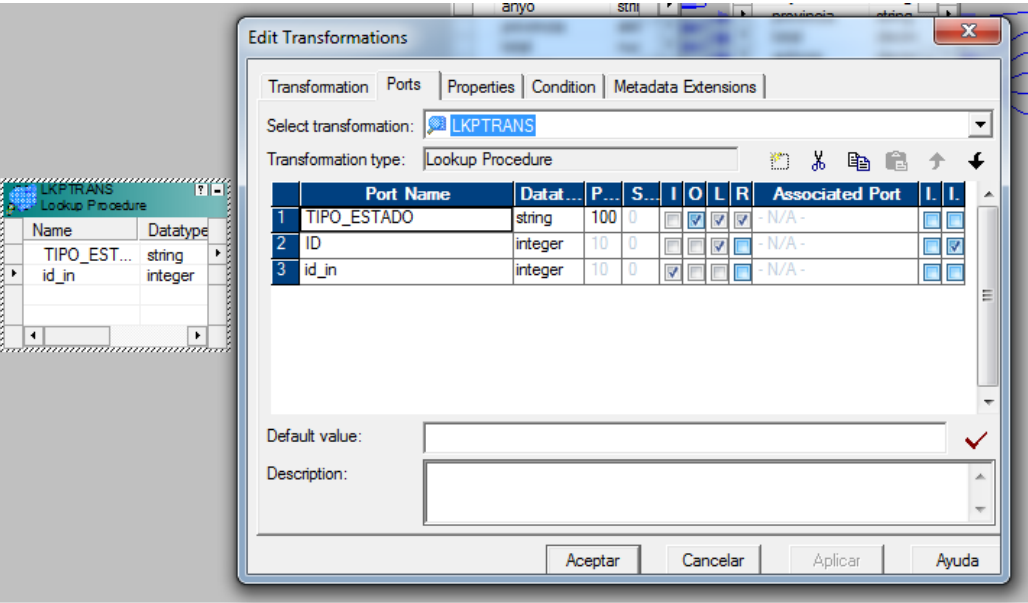


Ilustración 186: Configuración de campos de la lookup.

Como se muestra en la ilustración 57 la lookup tiene tres campos, los dos primeros corresponden a la tabla que vamos a consultar por eso tienen marcado el check con la letra L (lookup), además vemos que el primer campo tiene marcado el check de la “O” (output) y de la “R” (return) que indica que es el campo de salida y que se devuelve si la invocas desde un campo como una lookup desconectada.

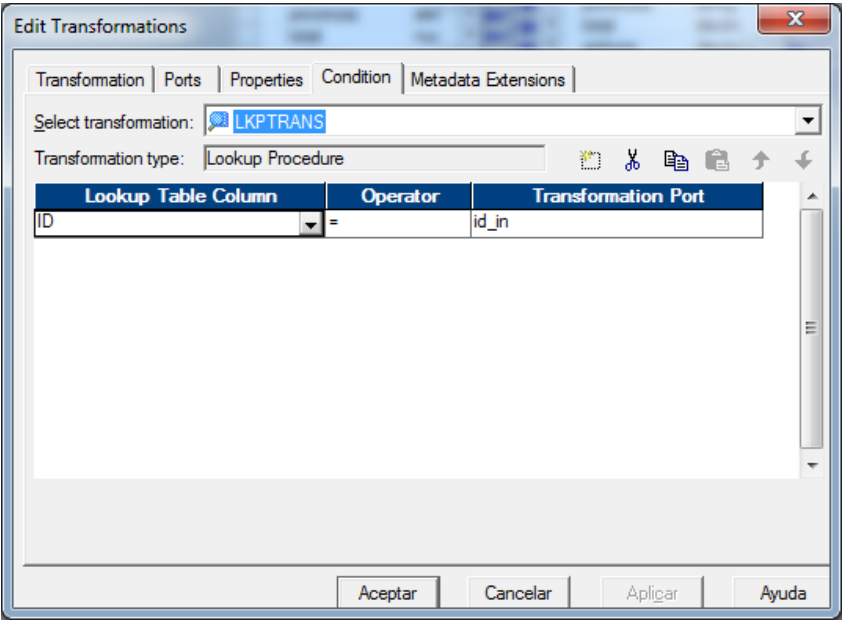


Ilustración 187: Filtro de condición de la lookup.

Por último podemos indicar que condición de filtrado queremos que aplique la lookup. En este caso comparamos el identificador de la tabla PFC\_TIPO\_ESTADO con el identificador con el que le invocamos desde la expresión anteriormente explicada. Y como hemos visto en la

configuración anterior nos devuelve el campo tipo\_estado en el cual coincidan sus identificadores.

Una vez que hemos obtenido los cuatro flujos y hemos obtenido el tipo de estado para cada rama, volvemos a unificar los flujos con un componente UNION previo a la realización del último cálculo y su inserción en tabla.

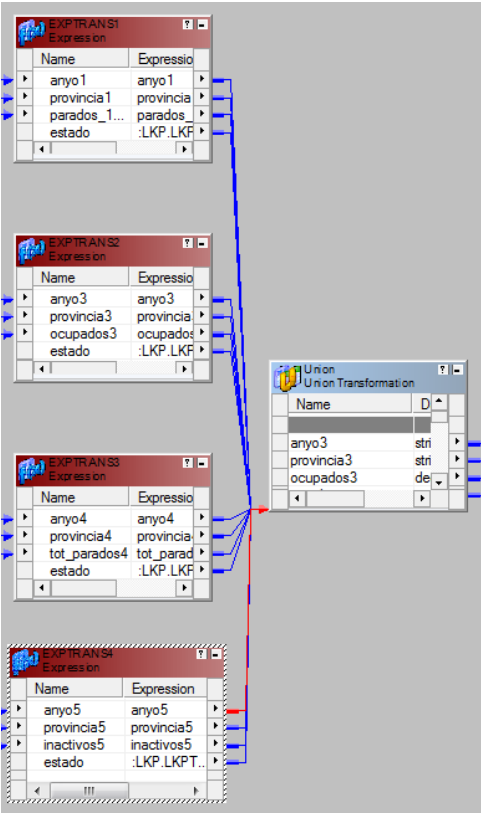
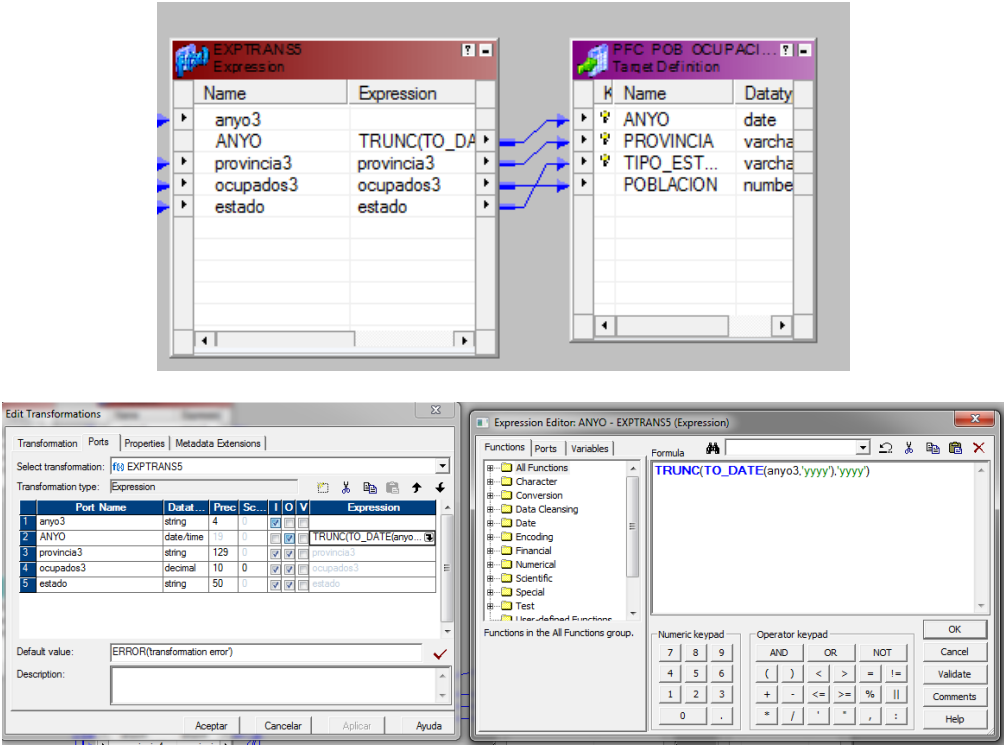


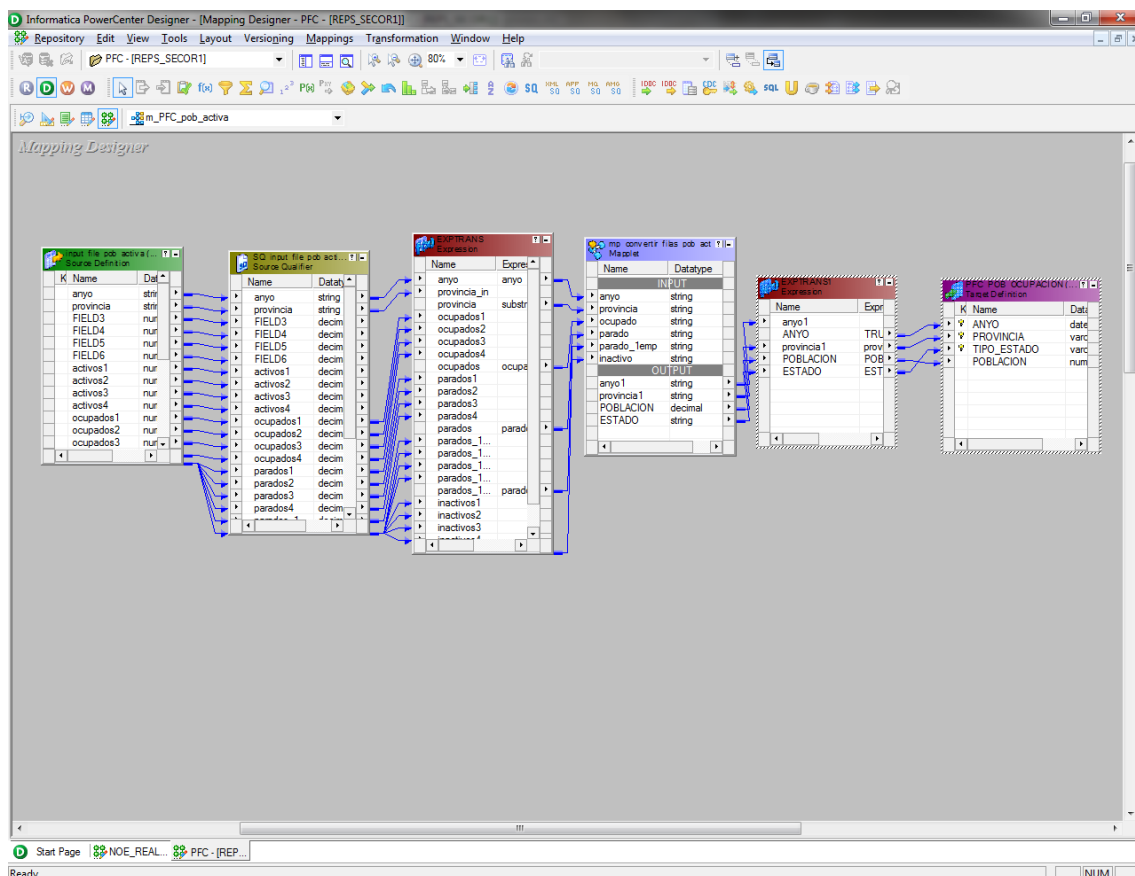
Ilustración 188: Unificación de flujos.

Una vez unificados los flujos, los registros pasan por una última expresión que cambia el tipo de datos de la fecha de una cadena (String) a un tipo fecha (date) y además le da el formato de año truncándola a año. Como se puede apreciar en la siguiente ilustración, en la configuración del cambio de tipo, para calcular el año se realiza la siguiente fórmula: “**TRUNC(TO\_DATE(**anyo3,**'yyyy')**),**'yyyy')**”. Aquí se realiza un cambio de tipo de datos con la expresión to\_date indicándole el formato de entrada de la cadena y después truncamos la fecha y le asignamos la mascara de año. Una vez realizado este cálculo se inserta en bbdd.



**Ilustración 189: Cálculo del año e inserción en tabla.**

Una vez explicado el proceso de carga para el año 2004 vamos a explicar el proceso de carga para el resto de años, desde el 2005 hasta el 2013. Este mapeo tiene distintas particularidades con respecto al anterior. En un primer momento estos ficheros de entrada tienen la información fraccionada por trimestres de un mismo año por lo que habrá que sumar dichos valores que están en distintos campos para poder obtener el valor anual que es el que estamos representando y el mismo que teníamos en el anterior fichero del 2004.



Para empezar realizamos la lectura de los ficheros de entrada y recogemos los registros leídos en una expresión inicial sobre la cual realizaremos la suma de los distintos campos que debido a su orden sabemos que contienen los datos que necesitamos para obtener los valores que buscamos. Este orden será el siguiente, empezaremos con los dos primeros campos que serán la clave, el año y la provincia, posteriormente tendremos 4 valores serán los correspondientes a los 4 trimestres del año que estemos procesando y que identificarán a la población ocupada, seguidamente los siguientes 4 valores corresponderán a la población parada, a continuación los siguientes 4 valores serán los parados en busca del primer empleo y por último estarán los 4 valores que sumaran el número de personas inactivas.

De esta manera la expresión quedaría de la siguiente manera que se muestra en la ilustración 62. Podemos ver como se ha calculado el campo provincia como una sub cadena en la cual hemos eliminado el código numérico que nos venia en el fichero de entrada, para ello hemos aplicado la siguiente fórmula “**substr**(provincia\_in,4)”.

Por otro lado para el calculo de los distintos campos sumatorios de los estados, se aplica la siguiente fórmula “**ocupados1+ocupados2+ocupados3+ocupados4**” es una suma de los distintos valores trimestrales del tipo de estado concreto que estemos tratando.

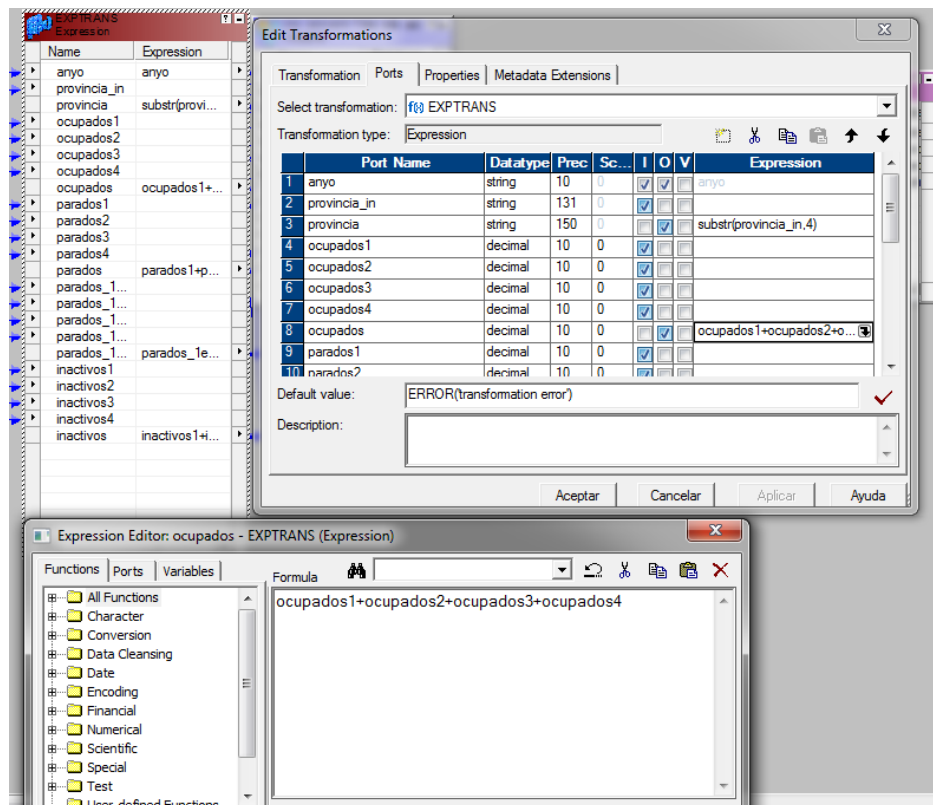


Ilustración 191: Configuración de la expresión inicial.

Una vez calculados estos campos se envían los registros con los nuevos campos a un mapplet que convertirá las distintas columnas de datos en filas distintas para cada tipo de datos realizando un producto cartesiano e insertándole el tipo de estado que le corresponda a cada valor.

La vista general del mapplet será la siguiente:

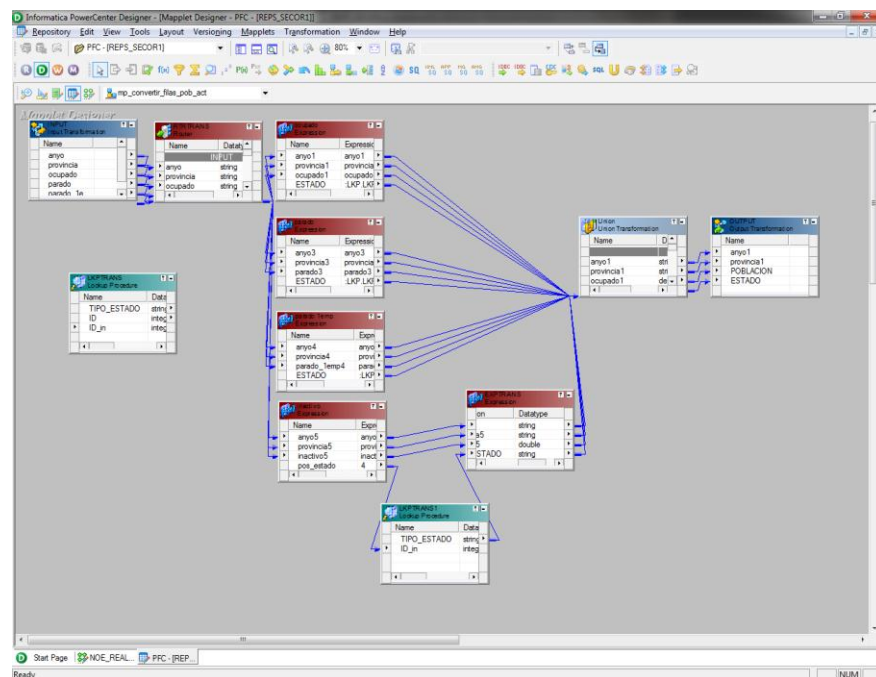
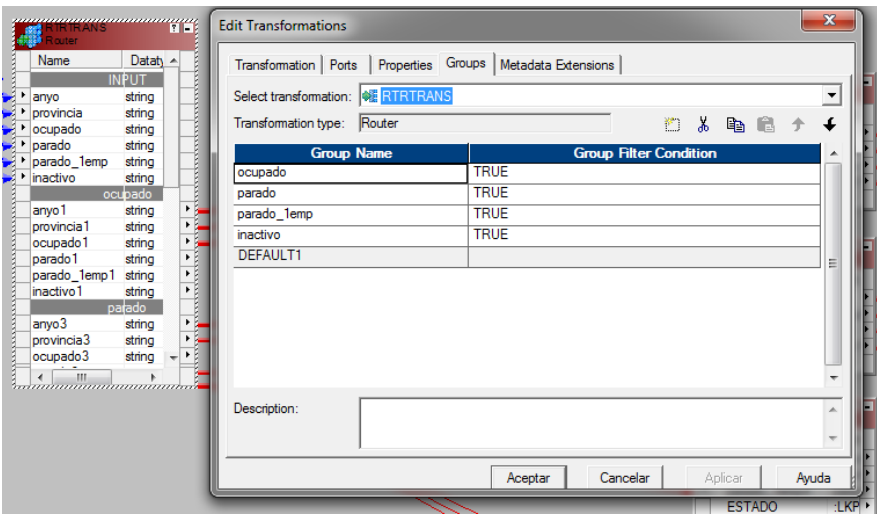


Ilustración 192: Vista general del mapplet mp\_convertir\_filas\_pob\_act.



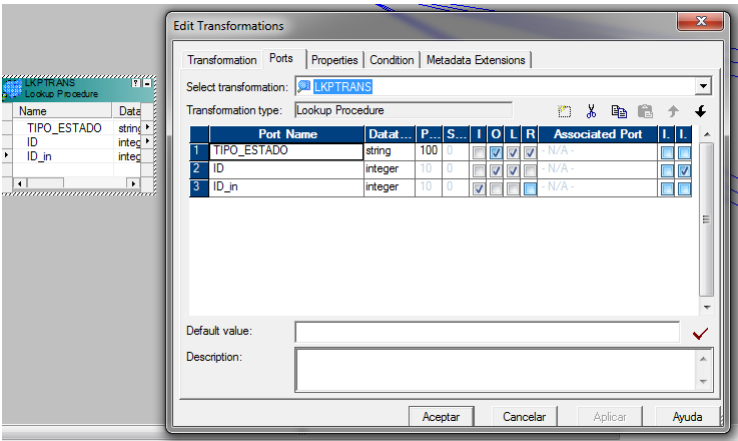
En primer lugar cuando el mapplet recibe los registros de entrada realiza un producto cartesiano de cada uno de ellos para dividir el registro que contiene los valores de los 4 tipos de estado ya mencionados en cuatro registros distintos para cada tipo de estado.



**Ilustración 193: Configuración del router para el tipo de estado.**

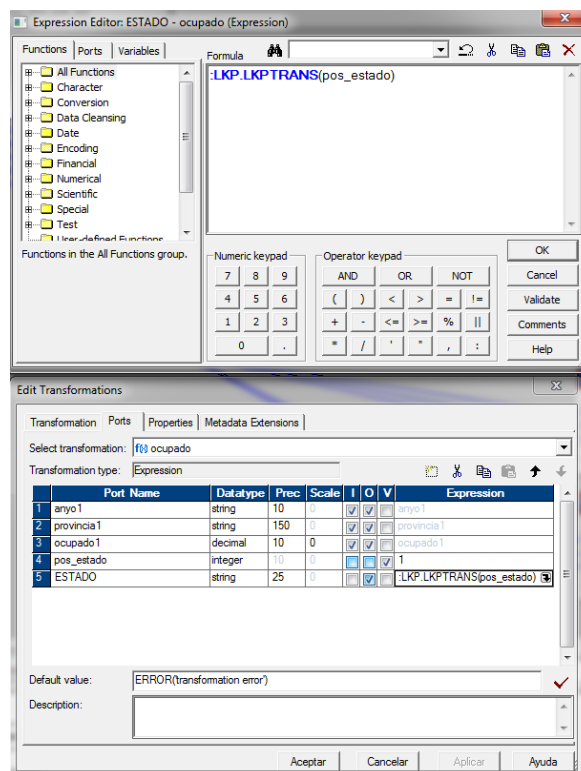
Una vez que tenemos los cuatro flujos ya separados para cada estado vemos como cada rama solo recoge los datos de los valores que desea calcular y los recoge una expresión que calculará el tipo de estado correspondiente ayudándose de una lookup al igual que hicimos en el anterior proceso para el año 2004. En este caso para realizar un ejemplo mas didáctico y poder explicar mas utilidades de la herramienta hemos realizado el calculo del tipo de estado con dos tipos de lookups distintas unas desconectadas y otra conectada para poder ver las diferencias.

El ejemplo de la lookup desconectada sería el siguiente, en un primer momento tenemos la lookup configurada de la siguiente manera, vemos como tenemos un identificador de entrada que será el que se le pase como parámetro en la llamada a la lookup, esto lo señalizamos marcando el check "I" que indica cuales son los campos de entrada y tenemos por otro lado los campos de la lookup y de salida que se indican marcando el check "O" y "L", además tenemos un único campo de retorno en el caso de las lookups desconectadas que en nuestro caso es el campo tipo de estado y que se señala marcando el check "R".



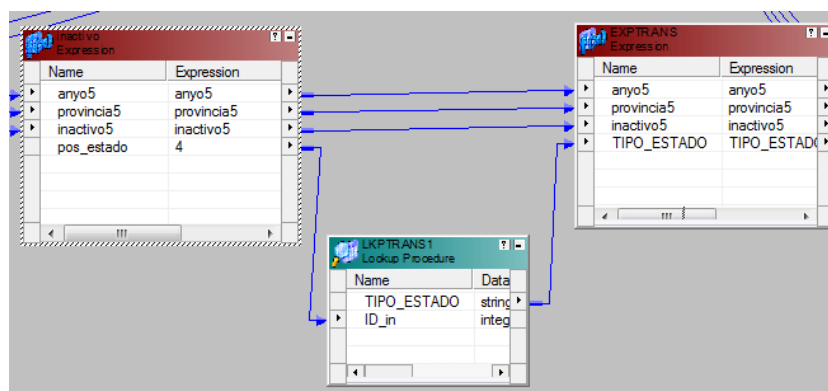
**Ilustración 194: Configuración lookup desconectada.**

A esta lookup se le aplica la condición o filtro de que ambos identificadores el de la tabla en la que se basa la lookup y el que le pasamos como parámetro sean iguales y se la llama desde el campo que estemos calculando en la expresión correspondiente.



**Ilustración 195: Invocación a la lookup desde la expresión.**

El otro ejemplo de lookup que tenemos en este mapplet corresponde a una lookup conectada que tiene el siguiente aspecto. Como podemos ver ahora una expresión se conecta con la lookup enviándole el valor del campo pos\_estado y la lookup esta configurada para comparar los identificadores y devolver como salida el campo tipo\_estado a la siguiente expresión.



**Ilustración 196: lookup conectada.**

La configuración de la lookup seria la que se muestra en la siguiente ilustración. Vemos como se le indica que tenemos un campo de entrada que es un identificador y como tenemos por parte de la lookup (la tabla que vamos a consultar) el campo ID que solo tiene marcado el check “L” que indica que será un campo que usaremos internamente en la lookup, en este caso se usara para realizar la condición de comparación pero que no será un campo de salida y por otro lado tenemos el campo TIPO\_ESTADO que vemos que tiene marcados los checks “O” y “L” que indica que de esta lookup saldrá el campo TIPO\_ESTADO perteneciente a la tabla a la que haga referencia la lookup, en este caso la tabla PFC\_TIPO\_ESTADO.

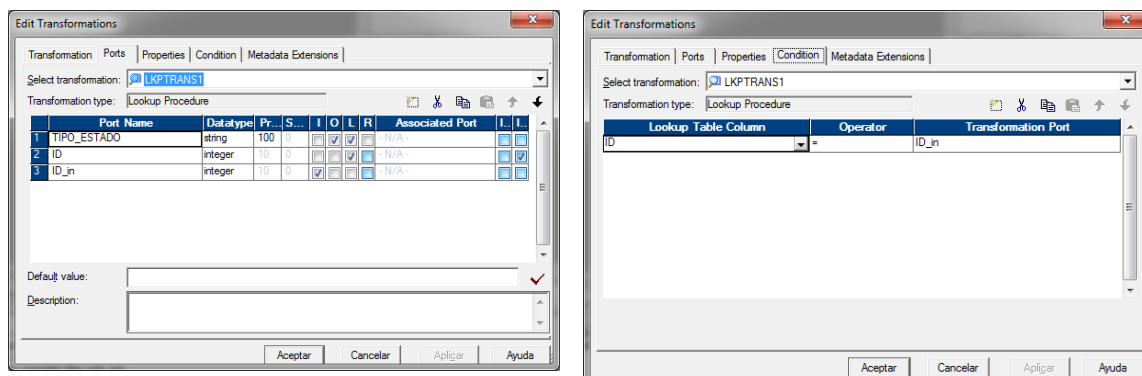


Ilustración 197: Configuración de la lookup conectada.

Una vez que hemos identificado en cada registro el tipo de estado de los valores que se están tratando se vuelven a unificar los 4 flujos en uno único mediante el uso de un operador UNION y de este irán al operador de salida del mapplet y se retomará el flujo en el mapping inicial.

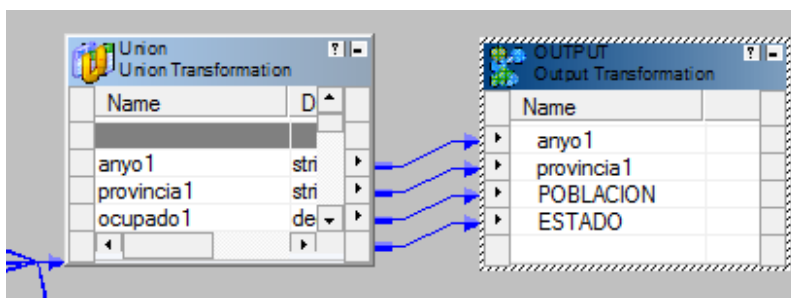


Ilustración 198: Unión de los flujos de datos y salida del mapplet.

Una vez retomado el flujo de datos en el mapeo inicial, los registros son recibidos por una expresión que realizará el calculo del año de la siguiente manera “TRUNC(TO\_DATE(anyo1,'YYYY'),'YYYY')”, realizamos el cambio de tipo de datos con la función to\_date indicándole que recibe una cadena de texto con solo el año y después truncamos la fecha para que la sitúe en el 1 de enero del año correspondiente.

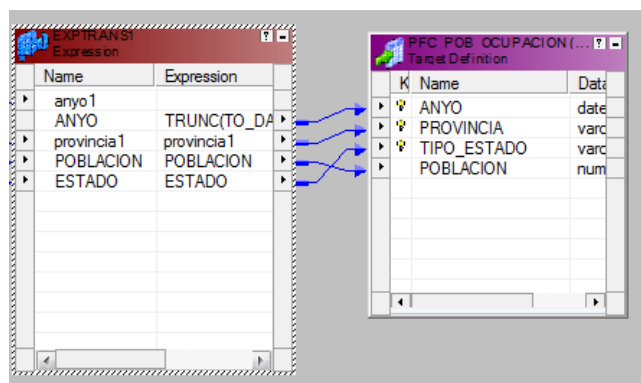


Ilustración 199: Expresión de cálculo del año e inserción en bbdd.

A nivel del workflow, en esta ocasión tenemos un workflow con dos sesiones en serie, esto implica que la segunda sesión no se lanza hasta que no finaliza la primera, al ser datos que no están relacionados, este proceso de carga podríamos configurarlo con sus sesiones en paralelo para así agilizar la carga y optimizar el rendimiento.

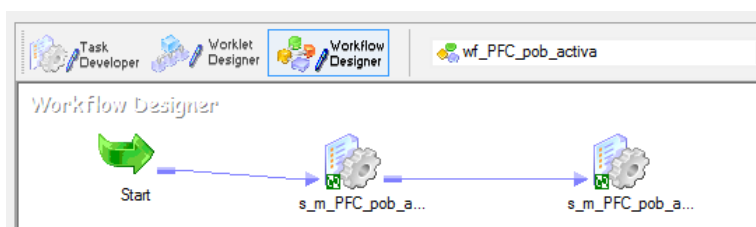


Ilustración 200: workflow con sesiones en serie.

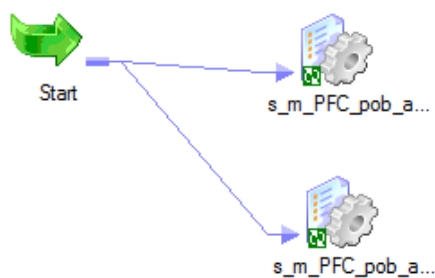


Ilustración 201: workflow con sesiones en paralelo.

Las sesiones se configuran de la misma manera que hemos visto hasta ahora salvo con la excepción, que en este caso tendremos que indicar en la pestaña de mapping también la conexión a la bbdd que utilizan las lookups. Describiremos únicamente la configuración de una sesión ya que ambas son iguales.

En un primer momento en la pestaña de properties configuraremos el directorio y el nombre que queremos que tenga el fichero de logs de ejecución.

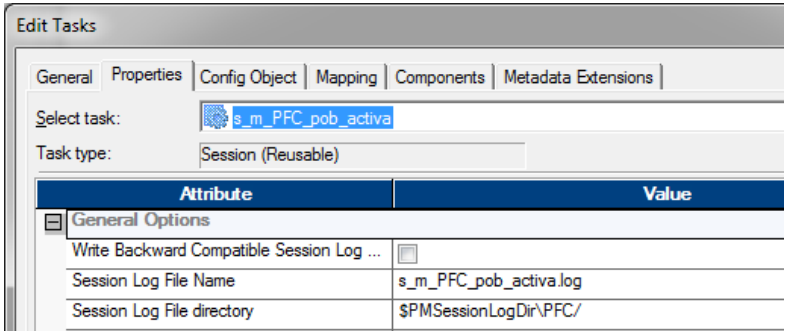


Ilustración 202: configuración de los logs de sesión.

Posteriormente en la pestaña de mapping tendremos que configurar las entradas en la cual indicaremos que tipo de entrada tenemos fichero o tabla en este caso será un fichero “File Reader” e indicaremos las características del fichero de entrada, indicaremos que se trata de un fichero indirecto (fichero que contiene una lista de referencias a otros ficheros que son los que contienen los datos) e igualmente le indicaremos cual es y donde esta situado este fichero de entrada.

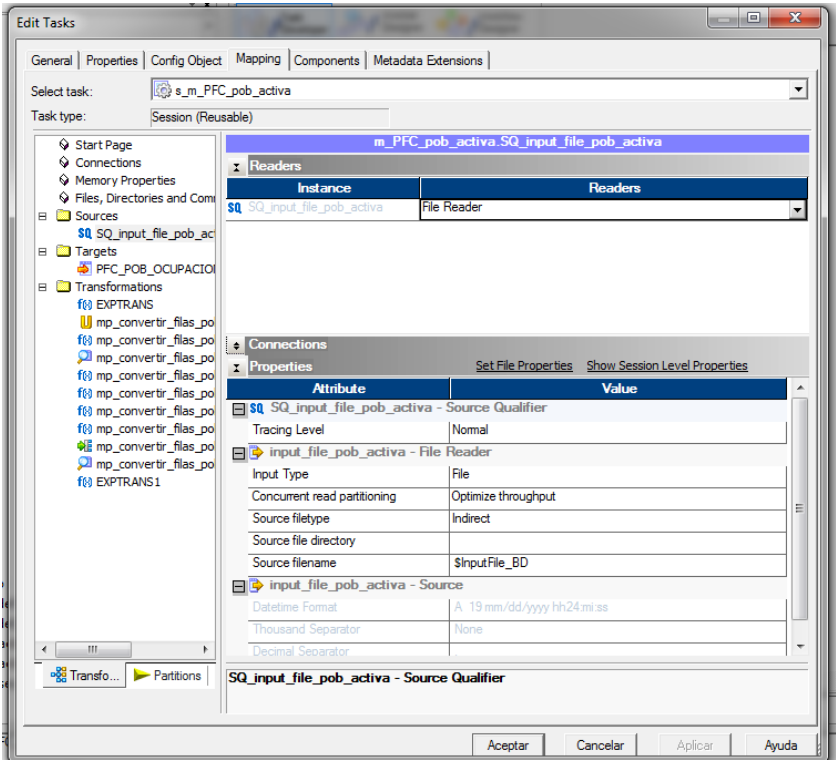


Ilustración 203: Configuración del fichero de entrada.

Para las salidas, en nuestro caso tendremos que indicar que tipo de salida usaremos, que será una tabla de bbdd y le informaremos de que conexión a bbdd tendrá que usar. Además indicamos el tipod e carga que queremos que realice (una carga normal, según llegan los registros se realiza el commit de los mismos) y le indicaremos el directorio y nombre del fichero donde queremos que deje los logs de los registros desechados.

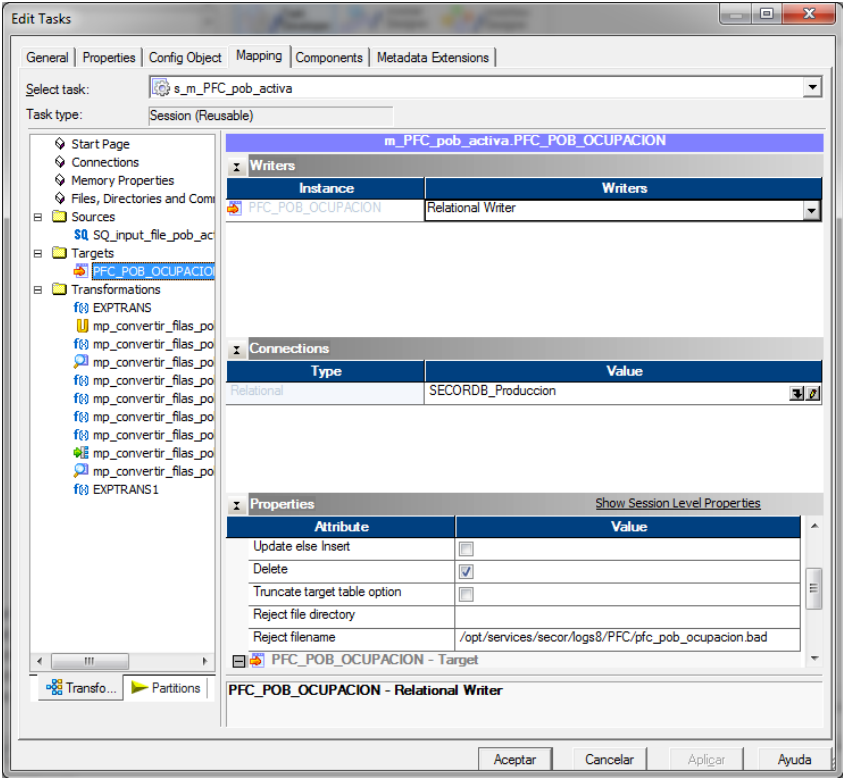


Ilustración 204: Configuración de las salidas.

Por último y dado que en nuestro mapplet tenemos dos lookups tenemos que indicar para cada lookup a que tabla tiene que consultar y que conexión a bbdd debe de usar para encontrarla.

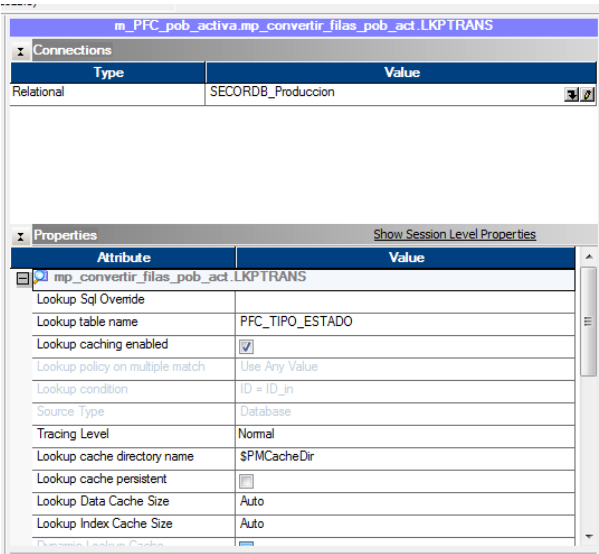


Ilustración 205: Configuración de la lookup.

### 3.5 Análisis de los datos e informes con POWERCENTER.

Ahora comentaremos los posibles estudios que se podrían realizar con los resultados obtenidos de las cargas anteriormente explicados.

Para ilustrar algunos ejemplos hemos utilizado el documento del INE (Instituto Nacional de Estadística) en el cual se muestra la forma de calcular algunos indicadores demográficos básicos. De igual manera intentaremos obtener algunos otros indicadores calculados por nosotros mismos atendiendo a los datos obtenidos.

#### 3.5.1 Indicadores concernientes a nacimientos.

En primer lugar hemos creado una vista para calcular la **tasa de natalidad bruta por provincia y municipio**. Esta se define como el total de nacimientos de madre con residencia en la provincia y el municipio  $i$  durante el año  $t$  por cada 1.000 habitantes. Es decir:

$$TBN^t(i) = \frac{N^t(i)}{P^t(i)} \cdot 1000$$

Ecuación 1: Tasa de natalidad bruta por provincia.

Donde:

$N^t(i)$  = Nacimientos de madre residente en la provincia  $i$  durante el año  $t$ .

$P^t(i)$  = Población media residente en la provincia  $i$  a lo largo del año  $t$ .

En el caso de la tasa de natalidad bruta por provincia para madres españolas hemos creado la vista `PFC_VW_TASA_NAT_BRUTA`, en ella podemos obtener como según la provincia y municipios de las mismas ha ido creciendo la natalidad durante los sucesivos años, de tal manera que podemos concluir si una provincia y municipio ha ido creciendo o decreciendo y en que porcentaje se ha ido creciendo.

En este caso podremos realizar los estudios atendiendo a los años que queramos estudiar a la provincia y a los municipios con un determinado rango de población.

Como ejemplo ilustrativo de este caso podemos mostrar la siguiente tabla:

PROVINCIA	MUNICIPIO	ANYO	TASA_NAT_BRUTA
Madrid	mas_50000	2003	4,4491
Madrid	mas_50000	2004	4,5237
Madrid	mas_50000	2005	4,4253
Madrid	mas_50000	2006	4,7106
Madrid	mas_50000	2007	5,0113
Madrid	mas_50000	2008	5,2655
Madrid	mas_50000	2009	4,9739
Madrid	mas_50000	2010	4,8947
Madrid	mas_50000	2011	4,7226
Madrid	mas_50000	2012	4,563

Tabla 32: Tasa de natalidad bruta por provincia y municipio.

De esta tabla se puede concluir que para la provincia de Madrid en los municipios mayores de 50000 habitantes el crecimiento de la natalidad ha sido bastante parejo durante los últimos 10 años no habiendo grandes diferencias de nacimientos en estos municipios de un año para otro. Aunque si se observa una tendencia de crecimiento hasta el 2008 y a partir de este año como la tasa va decreciendo a un ritmo parecido al del crecimiento anterior o incluso algo más acentuado.

Para la tasa de natalidad bruta por provincia para madres extranjeras hemos creado la vista PFC\_VW\_TASA\_NAT\_BRUTA\_EXT, en ella se calcula la tasa de natalidad al igual que en la vista anterior pero en este caso podemos diferenciar también por sexo, es decir podemos calcular la tasa de natalidad bruta para niños y niñas, esto nos permite ver por ejemplo si es mayor el número de nacimientos de niños que de niñas, también podemos si se suman estas estadísticas junto con las anteriormente mencionadas, tendríamos la tasa total de natalidad. En este caso en concreto no se ha hecho división por municipio ya que los datos originales no disponían de este nivel de detalle.

Para el ejemplo concreto que comentábamos anteriormente, en este caso las cifras obtenidas serían:

PROVINCIA	SEXO	ANYO	TASA_NAT_BRUTA_EXT
Madrid	Mujer	2003	3,363
Madrid	Hombre	2003	3,3782
Madrid	Mujer	2004	3,6823
Madrid	Hombre	2004	3,69
Madrid	Mujer	2005	3,7786
Madrid	Hombre	2005	3,7982
Madrid	Mujer	2006	4,1503
Madrid	Hombre	2006	4,1493
Madrid	Mujer	2007	4,8658
Madrid	Hombre	2007	4,8796
Madrid	Mujer	2008	5,202
Madrid	Hombre	2008	5,2101
Madrid	Mujer	2009	4,8136
Madrid	Hombre	2009	4,8263
Madrid	Mujer	2010	4,48
Madrid	Hombre	2010	4,5162
Madrid	Mujer	2011	4,0634
Madrid	Hombre	2011	4,0805
Madrid	Mujer	2012	3,8995
Madrid	Hombre	2012	3,8875

Tabla 33: Tasa de natalidad bruta para extranjeros.

Atendiendo a estos datos podemos ver como en el caso de los nacimientos para familias extranjeras en la provincia de Madrid como al igual que en los nacimientos nacionales la tendencia de nacimientos fue creciendo proporcionadamente hasta el año 2008 y como a partir de este momento ha ido decreciendo. También podemos observar como la tasa para varones y hembras ha sido similar no habiendo en ningún momento una tasa mucho mayor de nacimientos de un sexo que de otro.



Si lo comparamos con los nacimientos nacionales podemos observar como se comentaba, que se produjo un crecimiento demográfico hasta el año 2008 y a partir de este año ha ido disminuyendo la tasa de nacimientos, si podemos observar sin embargo que la tasa de nacimientos para madres extranjeras ha sido menor que para madres nacionales, esto puede deberse a que actualmente no haya tanta inmigración como pudiera haberla antes de la crisis actual.

Otro indicador que hemos calculado con respecto al número de nacimientos es el **ratio de masculinidad y feminidad por provincia**, estos cálculos solo los hemos podido realizar para los nacimientos extranjeros ya que es indispensable conocer el dato de los nacimientos por sexo para poder calcularlo y nosotros no teníamos estos datos para los nacimientos de madres nacionales.

El ratio de masculinidad y feminidad por provincia se define como el total de niños por cada 100 de nacimientos de niñas de madres residentes en España a lo largo del año t . Es decir:

$$RMN^t = \frac{N_{\text{varones}}^t}{N_{\text{mujeres}}^t} \cdot 100$$

**Ecuación 2: Ratio de masculinidad/feminidad al nacimiento.**

Donde:

$N_{\text{varones}}^t$  = Nacimientos de varones ocurridos durante el año t de madres extranjeras residentes en España.

$N_{\text{mujeres}}^t$  = Nacimientos de mujeres ocurridos durante el año t de madres extranjeras residentes en España.

En este caso hemos creado una vista llamada PFC\_VW\_RATIO\_MASC\_FEM\_NAC\_EXT, los datos que se muestran son de la siguiente manera:

PROVINCIA	ANYO	RATIO_MASCULINIDAD	RATIO_FEMINIDAD
Madrid	2003	100,4523	99,5497
Madrid	2004	100,2105	99,7899
Madrid	2005	100,5192	99,4835
Madrid	2006	99,9759	100,0241
Madrid	2007	100,2839	99,7169
Madrid	2008	100,1563	99,8439
Madrid	2009	100,2635	99,7372
Madrid	2010	100,8087	99,1978
Madrid	2011	100,4209	99,5808
Madrid	2012	99,6922	100,3088

Tabla 34: Ratio de masculinidad y feminidad.

Con estos datos podemos comprobar como por ejemplo para la provincia de Madrid el ratio de nacimientos de niños es mayor a lo largo de los últimos años que con respecto al numero de niñas, esto se cumple así salvo en los años 2006 y 2012 en cuyo caso el numero de nacimientos de niñas fue superior al de niños. También se puede ver como a pesar de estas cifras se puede concluir en que el numero de nacimientos de niños y niñas son muy similares y apenas hay diferencia entre unos valores y otros.

De igual manera estos datos confirman lo que vimos en el anterior indicador de la tasa bruta de nacimientos extranjeros en el cual vimos como las tasas eran muy parejas en sus resultados para los hombres y mujeres y como en los años 2012 y 2006 las tasas de nacimientos de las mujeres superaban a las de los hombres.

### 3.5.2 Indicadores concernientes a defunciones.

Para el caso de las defunciones vamos a calcular los mismos indicadores que para la natalidad ya que estos indicadores junto con los anteriormente explicados nos permitirían de manera fidedigna sacar conclusiones acerca de como han ido creciendo o decreciendo las poblaciones en España y podríamos confirmarlo con los datos de los censos de población para esos años. De esta manera podremos ver que poblaciones y provincias han ido creciendo o disminuyendo su población y como se distribuye la misma.

En primer lugar calcularemos el indicador de la **tasa de mortalidad bruta por provincia**, este indicador se define como el total de defunciones de residentes en la provincia  $j$  a lo largo del año  $t$  por cada 1.000 habitantes. Es decir:

$$TBM^t(j) = \frac{D^t(j)}{P^t(j)} \cdot 1000$$

Ecuación 3: Tasa de mortalidad bruta por provincia.

Donde:

$D^t(j)$  = Defunciones de residentes en la provincia  $j$  ocurridas durante el año  $t$ .

$P^t(j)$  = Población media residente en la provincia  $j$  a lo largo del año  $t$ .

En el caso de la tasa de defunción bruta por provincia para personas de nacionalidad española hemos creado la vista `PFC_VW_TASA_DEF_BRUTA`, en ella podemos obtener como según la provincia y municipios de las mismas ha ido creciendo la mortalidad a lo largo de los años, de tal manera que podemos concluir si una provincia y municipio ha ido creciendo o decreciendo y en que porcentaje se ha ido creciendo.

En este caso podremos realizar los estudios atendiendo a los años que queramos estudiar a la provincia y a los municipios con un determinado rango de población.

Como ejemplo ilustrativo de este caso podemos mostrar la siguiente tabla:

PROVINCIA	MUNICIPIO	ANYO	TASA_DEF_BRUTA
Madrid	mas_50000	2003	1,7004
Madrid	mas_50000	2004	1,6831
Madrid	mas_50000	2005	1,6638
Madrid	mas_50000	2006	1,6463
Madrid	mas_50000	2007	1,7681
Madrid	mas_50000	2008	1,7738
Madrid	mas_50000	2009	1,8323
Madrid	mas_50000	2010	1,8163
Madrid	mas_50000	2011	1,8281
Madrid	mas_50000	2012	1,9242

Tabla 35: Tasa de defunciones bruta por provincia y municipio.

En esta tabla podemos ver los datos para el caso concreto de la provincia de Madrid y los municipios con más de 50000 habitantes, estos datos complementarían a los anteriormente calculados de los nacimientos dándonos así una perspectiva más clara del crecimiento y

decrecimiento poblacional en la ciudad de Madrid con respecto a la población nacional en los últimos años.

Con estos datos concretamente podemos ver como durante los años 2004 al 2006 hubo un descenso de las defunciones en Madrid y como posteriormente se ha ido incrementando la mortalidad paulatinamente.

Si unificamos y estudiamos estos datos en conjunto con los de los nacimientos (Tabla 32: Tasa de natalidad bruta por provincia y municipio.), podemos ver como para Madrid en los municipios con mas de 50000 habitantes y refiriéndonos exclusivamente a las personas con nacionalidad española, entre los años 2004 y 2006 hubo un crecimiento de estas poblaciones debido a que se aumento con respecto a años anteriores el numero de nacimientos y se disminuyo el numero de defunciones en los mismos, sin embargo desde el 2009 en adelante, el crecimiento de la población esta disminuyendo ya que el numero de nacimientos esta disminuyendo lentamente y el numero de defunciones esta aumentando.

Para el estudio de los datos de defunciones de las personas con nacionalidad extranjera hemos creado la vista PFC\_VW\_TASA\_DEF\_BRUTA\_EXT, con ella de igual manera que en los nacimientos podemos ver como se distribuye la tasa de mortalidad a lo largo de los años para las provincias y el sexo elegido.

Por ejemplo podemos tener la siguiente tabla de datos:

PROVINCIA	SEXO	ANYO	TASA_DEF_BRUTA_EXT
Madrid	Hombre	2003	0,2277
Madrid	Hombre	2004	0,231
Madrid	Hombre	2005	0,2173
Madrid	Hombre	2006	0,2072
Madrid	Hombre	2007	0,2772
Madrid	Hombre	2008	0,2301
Madrid	Hombre	2009	0,2363
Madrid	Hombre	2010	0,2095
Madrid	Hombre	2011	0,2284
Madrid	Hombre	2012	0,2105
Madrid	Mujer	2003	0,2224
Madrid	Mujer	2004	0,2186
Madrid	Mujer	2005	0,2168
Madrid	Mujer	2006	0,1947
Madrid	Mujer	2007	0,2629
Madrid	Mujer	2008	0,2282
Madrid	Mujer	2009	0,2297
Madrid	Mujer	2010	0,2113
Madrid	Mujer	2011	0,221
Madrid	Mujer	2012	0,2073

Tabla 36: Tasa de defunción bruta de extranjeros por provincia.

Viendo esta tabla podemos ver como la tasa de mortalidad de las mujeres es siempre menor que la de los hombres esto puede deberse a que la esperanza de vida de las mujeres es mayor que la de los hombres. Podemos ver también como los valores de mortalidad no siguen una

progresión muy lineal en el tiempo si no que van dando saltos. Se puede ver como en el año 2007 con respecto a la población extranjera afincada en España hubo un gran aumento de las muertes con respecto a los años anteriores tanto en hombres como en mujeres y como después de este pico la progresión de la mortalidad ha ido disminuyendo.

Por último vamos a calcular el indicador del **Ratio de masculinidad y feminidad de defunciones extranjeras**, lo hacemos con los datos de las personas extranjeras ya que para las defunciones nacionales no disponemos de los datos que disciernan entre el sexo de los difuntos.

Este indicador se define como el número de defunciones de varones residentes en la provincia j por cada 100 defunciones de mujeres residentes en dicha provincia durante el año t. Es decir:

$$RMD^t(j) = \frac{D_{\text{varones}}^t(j)}{D_{\text{mujeres}}^t(j)} \cdot 100$$

**Ecuación 4: Ratio de masculinidad/feminidad a la defunción por provincia.**

Donde:

$D_{\text{varones}}^t(j)$  = Defunciones de varones residentes en la provincia j ocurridas durante el año t.

$D_{\text{mujeres}}^t(j)$  = Defunciones de mujeres residentes en la provincia j ocurridas durante el año t.

Para consultar estos datos hemos creado la vista PFC\_VW\_RATIO\_MASC\_FEM\_DEF\_EXT que muestra resultados con la siguiente estructura:

PROVINCIA	ANYO	RATIO_MASCULINIDAD_DEF	RATIO_FEMINIDAD_DEF
Madrid	2003	102,3585	97,6959
Madrid	2004	105,6738	94,6309
Madrid	2005	100,232	99,7685
Madrid	2006	106,4103	93,9759
Madrid	2007	105,4409	94,8399
Madrid	2008	100,8386	99,1684
Madrid	2009	102,863	97,2167
Madrid	2010	99,1209	100,8869
Madrid	2011	103,3473	96,7611
Madrid	2012	101,559	98,4649

Tabla 37: Ratio de masculinidad/feminidad a la defunción para extranjeros.

Con estos datos podemos comprobar como por ejemplo para la provincia de Madrid el ratio de defunciones de los hombres es mayor a lo largo de los últimos años que con respecto al numero de las mujeres, esto se cumple así salvo en 2010 en cuyo caso el numero de defunciones de mujeres fue superior al de hombres. También se puede ver como a pesar de estas cifras se puede concluir en que el número de defunciones de hombres y mujeres tiene bastantes diferencias, esto se debe probablemente a la esperanza de vida media en España, que en año 2012, según el INE (Instituto Nacional de Estadística) era de 79,4 años para los hombres y de 85,1 años para las mujeres.

De igual manera estos datos confirman lo que vimos en el anterior indicador de la tasa bruta de defunciones extranjeros en el cual vimos como las tasas eras dispares en sus resultados para los hombres y mujeres y como en el año 2010 la tasa de defunciones de las mujeres superaban a las de los hombres.

### 3.5.3 Indicadores concernientes a la población

Con los anteriores indicadores más los datos de la población que manejamos (censos, población activa y rentas por sector) podremos ver que poblaciones y provincias han ido creciendo o disminuyendo su población, como se distribuye la misma y si esta puede deberse a su tasa de desempleo o las condiciones salariales que hay en la zona.

En primer lugar vamos a calcular el indicador denominado **Saldo vegetativo por provincia**, este se define como la diferencia entre el total de nacimientos y el de defunciones de los residentes en la provincia  $j$  a lo largo del año  $t$ , por cada 1.000 habitantes. Es decir:

$$SV^t(j) = \frac{(N^t(j) - D^t(j))}{P^t(j)} \cdot 1000$$

Ecuación 5: Saldo vegetativo por provincia.

Un ejemplo de este indicador serían los datos obtenidos en la siguiente tabla, atendiendo a la provincia de Madrid y a los municipios de más de 50000 habitantes.

PROVINCIA	MUNICIPIO	ANYO	SALDO_VEGETATIVO
Madrid	mas_50000	2003	2,7487
Madrid	mas_50000	2004	2,8406
Madrid	mas_50000	2005	2,7615
Madrid	mas_50000	2006	3,0643
Madrid	mas_50000	2007	3,2432
Madrid	mas_50000	2008	3,4918
Madrid	mas_50000	2009	3,1416
Madrid	mas_50000	2010	3,0784
Madrid	mas_50000	2011	2,8945
Madrid	mas_50000	2012	2,6388

Tabla 38: Saldo vegetativo.

A raíz de estos datos podemos ver como en Madrid para los municipios de mas de 50000 habitantes el numero de nacimientos con respecto al numero de defunciones ha ido creciendo hasta el año 2008 y como a partir de ahí ha comenzado a decaer, lo que implica que el numero de nacimientos esta siendo menor proporcionalmente que el numero de muertes.

Otra comparación que podemos realizar con los datos obtenidos es la del **numero de parados comparado con el numero de nacimientos**, de tal manera que podremos observar si existe algún tipo de relación entre el número de nacimientos y el número de parados, para así comprobar si la situación actual de desempleo influye en el crecimiento del país.

Para ello hemos creado la vista PFC\_VW\_NACIMIENTOS\_PARADOS, esta nos devuelve datos del número de parados (teniendo en cuenta los activos parados y los activos parados que buscan su primer empleo) con el siguiente formato:

PROVINCIA	ANYO	NUM_NACIMIENTOS	NUM_PARADOS
Madrid	2004	69027	234300
Madrid	2005	69367	943800
Madrid	2006	71912	899600
Madrid	2007	74837	913400
Madrid	2008	78792	1251800
Madrid	2009	75957	2044900
Madrid	2010	73878	2397900
Madrid	2011	72023	2474100
Madrid	2012	69374	2799700

Tabla 39: Relación de parados y número de nacimientos.

Con estos datos podemos comprobar como el numero de nacimientos iba incrementándose a lo largo de los años en la provincia de Madrid hasta el año 2008 mientras las cifras de parados eras más o menos estables, aunque en incremento constante tenían un crecimiento lento. A partir del año 2009 cuando estalló la crisis en España y el número de parados de ese año con respecto al 2008 se duplicó prácticamente el número de nacimientos fue decreciendo de nuevo paulatinamente hasta llegar en 2012 a cifras de nacimientos similares a las del 2004.



Podemos concluir a tenor de estos datos que la crisis actual y el cauce de los acontecimientos influyen en la idea de las personas de formar familias o aumentarlas y que esto hace que el crecimiento demográfico de la provincia disminuya.

Por último vamos a intentar ver la **relación salarial entre hombres y mujeres por comunidad y sector**, de esta relación podremos ver la diferencia salarial que hay entre hombre y mujeres por sectores y comunidades y ver si estas diferencias han ido disminuyendo o acrecentándose a lo largo del tiempo. Los datos que vamos a obtener serán de la siguiente manera:

ANYO	COMUNIDAD	SECTOR	RATIO_SALARIO_HOM_MUJ
2004	Madrid (Comunidad de)	Construcción	-1,12
2005	Madrid (Comunidad de)	Construcción	-1,12
2006	Madrid (Comunidad de)	Construcción	-1,12
2007	Madrid (Comunidad de)	Construcción	-1,11
2008	Madrid Comunidad de	Construcción	-1
2009	Madrid Comunidad de	Construcción	-1,05
2010	Madrid Comunidad de	Construcción	-1
2011	Madrid Comunidad de	Construcción	-1,01
2004	Madrid (Comunidad de)	Industria	1,35
2005	Madrid (Comunidad de)	Industria	1,38
2006	Madrid (Comunidad de)	Industria	1,24
2007	Madrid (Comunidad de)	Industria	1,28
2008	Madrid Comunidad de	Industria	1,16
2009	Madrid Comunidad de	Industria	1,17
2010	Madrid Comunidad de	Industria	1,12
2011	Madrid Comunidad de	Industria	1,15
2004	Madrid (Comunidad de)	Servicios	1,54
2005	Madrid (Comunidad de)	Servicios	1,54
2006	Madrid (Comunidad de)	Servicios	1,48
2007	Madrid (Comunidad de)	Servicios	1,46
2008	Madrid Comunidad de	Servicios	1,34
2009	Madrid Comunidad de	Servicios	1,37
2010	Madrid Comunidad de	Servicios	1,31
2011	Madrid Comunidad de	Servicios	1,3

**Tabla 40: Relación salarial entre hombres y mujeres por comunidad y sector.**

Para obtener estos datos hemos creado la vista PFC\_VW\_RATIO\_SALARIO\_HOM\_MUJ, como podemos ver en los resultados que hemos obtenido en el sector construcción nos salen valores negativos esto se debe a que el número de observaciones muestrales para algún sexo está comprendido entre 100 y 500, por lo que la cifra es poco significativa y lo han señalado poniendo un valor negativo. En este caso indicaría que en sector construcción habría muy pocas mujeres en relación con el número de hombres, de todas formas viendo los valores absolutos podemos ver como en este caso los hombres cobraban mucho mas que las mujeres en torno al 12% más y como a partir del 2008 que estalló la crisis del ladrillo y que el sector construcción se debilitó mucho los salarios se han ido igualando.

Con respecto al sector industria podemos ver como hay una gran diferencia salarial entre hombres y mujeres, pero se puede observar como a lo largo de los años esta diferencia se ha

ido acortando pasando de un 35 o 38% en los años 2004 y 2005 a un 12 o 15% entre los años 2010 y 2011.

Por último con respecto al sector servicios podemos ver como es el sector en que mas discriminación entre hombres y mujeres hay ya que los datos comienzan con un 54% de diferencia salarial en el 2004 para los hombres y a pesar de que la tendencia ha sido la de ir disminuyendo esta diferencia aun en el 2011 la diferencia en este sector estaba en el 30%.

## 4 Conclusiones.

Finalmente, en este apartado se muestran conclusiones generales sobre el PFC y posibles líneas de futuro.

### 4.1 Conclusiones generales.

El objetivo principal de este proyecto es mostrar las principales funcionalidades de la herramienta Powercenter para realizar de manera eficiente la carga de datos en un almacén. La carga de datos es una de las operaciones más delicadas a la hora de crear y mantener un almacén de datos. Es una tarea compleja, no solo porque las fuentes de datos son heterogéneas, que ya tiene su complejidad, sino también hay que realizar un pre-procesado de datos transformándolos en datos que puedan ser integrados y analizados en el almacén.

Al ser una de las fases iniciales del diseño e implementación de un almacén de datos, es una de las tareas más delicadas, todos los errores cometidos y no analizados en esta fase serán arrastrados a las fases posteriores, lo que conllevará que el almacén no contenga la información necesaria para su buen funcionamiento y por lo tanto, la información que se extraiga del mismo no será del todo veraz.

Powercenter es una herramienta que proporciona funcionalidades necesarias para crear los procedimientos necesarios para realizar la carga de un almacén.

Este proyecto por lo tanto presentamos esta herramienta explicando el funcionamiento de las ETLs y la potencia que pueden tener estas, para el procesado y transformación de los datos de cara a su explotación.

A partir de los distintos ejemplos que hemos ido realizando y explicando vemos como Powercenter es una herramienta muy potente para el manejo de los datos de distintos entornos y fuentes, así de esta manera podemos, a partir de unos datos sin relación aparente y gracias a sus transformaciones, podemos finalmente realizar una mejor explotación de los datos y obtener resultados que a priori no parecían estar ahí.

En mi opinión queda demostrado que las ETLs y en concreto, Powercenter son una herramienta versátil y fácil de usar que potencia mucho el manejo y rendimiento del trabajo con la BBDD. En los ejemplos expuestos en este documento queda plasmado como gracias al uso de estas herramientas podemos potenciar nuestros informes de BI pudiendo realizar informes más sencillos y rápidos para obtener mejores conclusiones a partir de un tratamiento previo de los datos antes de su inserción en BBDD desde sus distintos orígenes.

Se ve como estas herramientas son sencillas e intuitivas lo que facilita su implantación en cualquier proyecto dándole a éste un valor añadido sobre otros proyectos que no lo tengan y hagan que la explotación de los datos pueda resultar más costosa.

Por último gracias a mi experiencia laboral trabajando con Powercenter puedo asegurar que el uso de estas herramientas ayudan mucho en los proyectos con un gran volumen de datos ya que de no usarse éstas se tendría que realizar la carga de los datos y su pre procesado con otros lenguajes de programación como pudiera ser java, estos procesados y cargas son mucho mas lentos y complicados de desarrollar, sin tener en cuenta su desarrollo que es mas laborioso y lento. Por el contrario el problema de esta herramienta en concreto es que es una

herramienta de pago y es muy cara, pero para solucionar este problema podemos usar distribuciones de otras ETLs que son libres, ahora mismo una de las ETLs de libre distribución más usadas es TALEND.

Debido a todos estos puntos pienso que las ETLs son actualmente unas herramientas muy a tener en cuenta en cualquier proyecto con un gran volumen de datos como los data warehouse y otras bbdd relacionales o incluso no sql, ya que estas herramientas ya tienen servicios que se integran con este tipo de BBDD.

## 4.2 Líneas de futuro.

A colación con el último punto de las conclusiones una posible línea de futuro podría ser realizar un ejemplo de integración con alguna base de datos no SQL.

También se podría utilizar algún tipo de herramienta de reporting para dar una perspectiva más global de un posible proyecto y como se pueden explotar los datos de una manera más eficiente y sencilla permitiendo así sacar mejores conclusiones.

De igual manera se puede realizar esta misma carga con otra herramienta y poder comprobar la eficiencia y versatilidad de ambas herramientas.

## 4.3 Bibliografía.

A lo largo del presente documento se han usado recursos obtenidos mayoritariamente de la ayuda de la aplicación y que me han ayudado a redactar y documentar muchos apartados.

A continuación se expone la lista de las referencias más representativas.

### Aplicación:

- Ayuda de la aplicación POWERCENTER.

### Fuentes electrónicas:

- Ficheros de entrada: <http://www.ine.es/>
- Web de la empresa Informática, de aquí obtuvimos el precio de las licencias de Powercenter:  
<http://www.informatica.com/es/products/data-integration/enterprise/powercenter/>
- Web de la empresa Oracle, en ella obtuvimos el precio de las licencias de la bbdd de nuestro estudio: <http://www.oracle.com/es/corporate/pricing/index.html>

### Documentos electrónicos:

- Indicadores demográficos básicos (obtenida del INE): “metodologia\_idb.pfd”.
- Almacenes de datos (Wikipedia).
- BBDD multidimensionales y OLAP (Wikipedia).

### Libros:

- Guía del aprendiz de Powercenter (empresa informática).



## 5 Gestión de proyectos.

En este apartado se describe el proceso que se ha seguido para desarrollar este trabajo. Como todo proyecto, se requiere una gestión adecuada. Se debe realizar una planificación donde se divida el trabajo a realizar en tareas y subtareas para que el desarrollo del proyecto se produzca de forma organizada.

Además se incluye una estimación del presupuesto necesario para desarrollar este proyecto.

### 5.1 Planificación.

De cara a la realización de un proyecto de este tipo hay que especificar claramente las tareas que se van a desarrollar y estimar el tiempo que nos llevará cada una de ellas. Habrá casos en los que se puedan realizar varias tareas en paralelo y otros en los que unas tareas dependerán de otras y por lo tanto no puedan realizarse en paralelo. Para ello hay que tener claro los aspectos y funcionalidad del proyecto que vamos a realizar y ver que desarrollos se podrán realizar o no en paralelo.

En nuestra planificación dado que es un proyecto pequeño y en solitario no pondré ninguna tarea en paralelo pero si que las iré fraccionando para que se vea el flujo de tareas que he ido siguiendo. Por otro lado consideraremos la duración como días de 8 horas y se apreciará como la tarea de documentación se ha realizado en paralelo con el desarrollo para que no se olvidara nada y por ello habrá días en los que

Una posible planificación para este desarrollo aquí explicado sería:

Tarea	Fecha inicio	Fecha fin	Duración
Análisis	08/01/2014		3
Desarrollo de scripts			4
Desarrollo de SQL			2
Desarrollo Powercenter			8
Pruebas			3
Documentación	08/01/2014		60
Implantación			1

Tabla 41: Lista de tareas.

La duración de las fases del proyecto son tiempos reales invertidos en la realización del proyecto, las fechas no coinciden con la realidad ya que he supuesto que sólo dedicara mi tiempo al PFC como una jornada laboral real, mientras que en mi caso he tenido que simultanear el proyecto con mi trabajo real por lo que las fechas de inicio y fin no coincidirán con la fecha real en la cual acabe el PFC ya que habría días en los que no habría invertido nada de tiempo.

He puesto la tarea de implantación como si fuera un proyecto real ya que esta sería el último paso en un proyecto, aunque en el caso de este tipo de procesos con ETL, este tiempo es muy reducido para un ejemplo tan pequeño como el que hemos tratado.

La duración total del proyecto ha supuesto 81 días que son 648 horas de trabajo, cargadas dichas horas como se puede ver principalmente a la documentación de este proyecto.

5.1.1 Diagrama de Gantt.

Para facilitar la comprensión y alcance temporal del proyecto hemos realizado el diagrama de Gantt para especificar el orden y tiempo de duración de cada tarea:

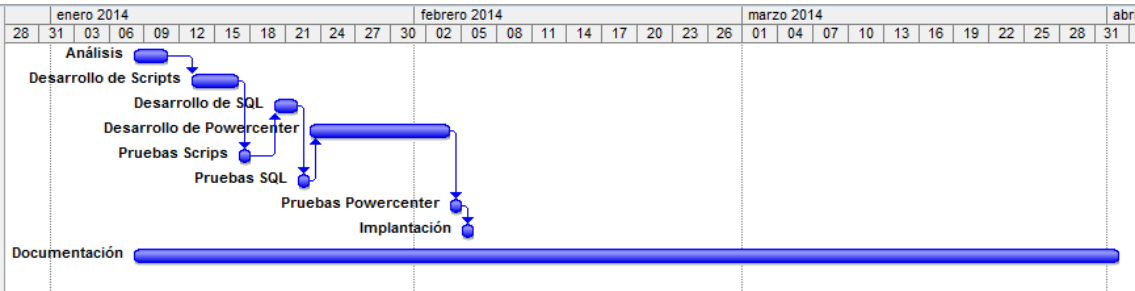


Ilustración 206: Diagrama de Gantt.

Cada tarea tiene a su izquierda el nombre que la identifica.

5.2 Presupuesto del proyecto.

Este apartado intentará dar una aproximación de un presupuesto para realizar un proyecto de este tipo. Habría que indicar que para este tipo de proyectos como el que se indica en este documento no sería nada aconsejable el uso de la ETL Powercenter ya que es una ETL de pago y sus licencias se venden por núcleo de procesador y son muy caras. Para este tipo de proyectos sería más aconsejable realizarlos con una ETL open source como pudiera ser TALEND.

Contamos con el hándicap en este apartado de que los costes reales de las licencias de Powercenter no son públicos, por lo que haremos una estimación según lo que he oído en los proyectos en los que he estado trabajando, de igual manera los equipos necesarios para montar este tipo de proyectos son equipos muy potentes y muy caros para poder almacenar las distintas BBDD y con los requerimientos de procesamiento y almacenamiento que se requiere.

Dividiremos el presupuesto atendiendo a tres recursos diferentes: hardware, software y recursos humanos necesarios.

### 5.2.1 Recursos Hardware.

Consideramos en este proyecto dos escenarios un para la parte del servidor (no sé exactamente las especificaciones técnicas del sistema montado) y otro para la parte del cliente. Los requerimientos serían los siguientes:

#### **En la parte servidora tendríamos:**

- Equipo sobre el que se instala la aplicación de power center y la bbdd, es un sun4u sparcs SUNW,Sun-Fire-V440.
- Conexión a internet punto a punto.

#### **En la parte cliente tendremos:**

- Equipo sobre el que se instala la aplicación cliente, que tiene las siguientes características:
  - Procesador Intel core i5 3.1 GHz.
  - 4 GB de memoria RAM.
  - Disco duro de 500 GB.
  - Tarjeta de sonido.
  - Tarjeta de video.
  - Tarjeta de red.
  - Teclado y ratón.
  - Monitor.
- Conexión punto a punto.

Estos son los elementos sobre los que se ha desarrollado la prueba de concepto, no es necesario tener exactamente el mismo equipamiento, también se han realizado partes sobre otros ordenadores de menor potencia y ha funcionado correctamente.



### 5.2.2 Recursos Software.

El software para este proyecto se podría dividir en la parte servidora y la parte cliente ya que esta herramienta es una herramienta tipo cliente-servidor. Voy a describir el sistema tal cual lo tenemos en el proyecto donde trabajo, es seguro que se pudiera realizar una configuración del sistema más barata y distinta a la que describamos en este apartado.

#### **En la parte servidora tendríamos:**

- Sistema operativo Unix: Solaris.
- BBDD: Oracle 11 g.
- ETL: Powercenter 8.1 sp5.

#### **En la parte cliente tendremos:**

- Sistema operativo Microsoft Windows 7 Professional Edition® de 64 bits. Sobre este sistema operativo se ejecutará tanto la aplicación final como los programas necesarios para desarrollarla.
- Cliente ftp Putty: Es un cliente de conexiones ssh y sftp a distintas maquinas.
- Cliente de bbdd, SQL developer: Es un cliente de conexión a bbdd Oracle.
- Cliente ETL, Powercenter 8.1 sp5: Herramienta para la realización de los procesos de carga a bbdd.
- Editor de texto, Ultraedit: editor de texto que se utiliza para la creación de los scripts de pre procesado.

### 5.2.3 Recursos humanos.

Para este proyecto podría realizar todas las labores una única persona con los conocimientos de análisis y de programación necesarios, no obstante de realizar una distinción de roles tradicional tendríamos los siguientes recursos.

Puesto	Precio/Jornada
<b>Jefe de proyecto</b>	160 €/día
<b>Analista</b>	192€/día
<b>Analista-Programador</b>	240 €/día

Tabla 42: Coste de recursos humanos.

- **Jefe de proyecto:** Es el máximo responsable del proyecto. Se encarga de la labor de planificación, seguimiento y estimación y utilización de los recursos humanos disponibles. Además es la cara visible con el cliente, manteniendo reuniones periódicas para el análisis y seguimiento del devenir del proyecto, garantizando los márgenes de calidad y rentabilidad establecidos.
- **Analista:** Responsable del estudio de los requisitos por parte del cliente y elaboración de los documentos técnicos y funcionales para el cliente y desarrolladores.
- **Analista-programador:** Es la persona que se encargará de interpretar los documentos técnicos y llevarlos acabo realizando su desarrollo correspondiente.

Además podríamos tener algunos otros roles de calidad, pruebas y documentación del desarrollo pero para este caso asumimos que recaerían sobre el analista-programador y el funcional.

#### 5.2.4 Coste del proyecto.

Vamos a estimar la dedicación en días de cada uno de estos roles sobre el tiempo total estimado para el desarrollo, que fueron 81 días, con jornadas de 8 horas diarias.

En algunos roles utilizaremos una carga que no será el 100%, esto implica que dicha persona le dedicará solo parte de la jornada a las labores de este proyecto y el resto de su jornada a otras labores en otros proyectos. De esta manera podemos alargar las jornadas estimadas para ese rol a lo largo del tiempo y usando menos jornadas tener un jefe de proyecto durante todo el proceso del proyecto, ya que cuando no haya una tarea específica para él no estará cargando horas sobre este proyecto, si no que estará gestionando otros.

Para este proyecto en concreto la labor del jefe de proyecto no se considera necesaria en un 100% de su tiempo ya que llevará a cabo la planificación inicial y el seguimiento de los hitos o tareas conseguidas. Para ello estimamos que no le será necesario más de 2 jornadas de trabajo que dado que se le necesitará de principio a fin del proyecto le asignaremos una carga o dedicación al proyecto de un 10% del tiempo de su jornada cuando este con el proyecto.

En el caso del análisis es un factor fundamental e imprescindible el hecho de conocer exactamente los requisitos del cliente y con que elementos vamos a contar. Para ello es necesario tener un buen conocimiento del negocio que se está tratando por parte del cliente y que el analista sea capaz de plasmar clara y concisamente en los documentos funcionales y técnicos lo que posteriormente va a implementar el analista programador. Por este motivo, hemos estimado unos 5 días de trabajo al 100% de carga para el analista en los cuales su labor será necesaria en dos partes. Al inicio del proyecto para la toma de requisitos y elaboración de la documentación técnica y al final de proyecto para la ayuda al analista en la elaboración de la documentación del desarrollo y memoria.

Por último el analista programador tendrá la doble labor de analizar e interpretar el documento técnico proporcionado por el analista y desarrollar lo que en ellos se trasmita. Además para este proyecto tan pequeño le hemos asignado la labor de pruebas y documentación que en otros proyectos mas grandes podría haber dos roles concretos para estas labores.

Con estas especificaciones los costes de los recursos humanos, con los tres roles anteriormente especificados serian:

Rol	Porcentaje dedicación	Jornadas dedicadas	Precio / jornada	Coste
Jefe de proyecto	0.1	2	240 €/día	480 €
Analista	1	5	192 €/día	960 €
Analista-programador	1	77	160 €/día	12320 €

Tabla 43: Tabla de coste de recursos humanos.

Con respecto al coste de material hardware (solo explicaremos la parte cliente ya que no sabemos exactamente que servidores tienen ni su posible precio) sería el siguiente:

Elemento	Precio
PC cliente	600 €

Tabla 44: Tabla de costes de hardware.

Con respecto a los costes de software, quitando los que puedan ser usados por licencias gratuitas serían estos:

Elemento	Precio
Solaris 10	1584 €
Oracle 11 g	34623 € (por procesador)
Powercenter 8.1	150000 € (por procesador)

Tabla 45: Tabla de costes de software.

El precio de estos elementos se vende en relación al número de procesadores que tenga el servidor, nosotros supondremos para cada caso que estaremos hablando de un único procesador.

En el caso de la licencia de Powercenter he realizado una estimación según los datos que tengo al respecto de los proyectos en los que he estado.

Para el caso del sistema operativo Solaris 10 el precio de la licencia por procesador para un año, ya que esta se vende por tiempo es el anteriormente indicado.

Por último añadiremos una provisión de 300€ para consumibles, posibles costes de desplazamiento a reuniones, etc. Todo esto hace que el coste total del proyecto presupuestado sea:

**COSTE FINAL PRESUPUESTADO:**

**200.867 €**